User Manual

MTS4EA
Compressed Video ES Analyzer

071-1641-02

This document supports software version 5.0.

www.tektronix.com

Contacting Tektronix

Tektronix, Inc.
14200 SW Karl Braun Drive
P.O. Box 500
Beaverton, OR 97077
USA

For product information, sales, service, and technical support:
- In North America, call 1-800-833-9200.
- Worldwide, visit www.tektronix.com to find contacts in your area.

Warranty 9(b)

Tektronix warrants that the media on which this software product is furnished and the encoding of the programs on the media will be free from defects in materials and workmanship for a period of three (3) months from the date of shipment. If any such medium or encoding proves defective during the warranty period, Tektronix will provide a replacement in exchange for the defective medium. Except as to the media on which this software product is furnished, this software product is provided "as is" without warranty of any kind, either express or implied. Tektronix does not warrant that the functions contained in this software product will meet Customer's requirements or that the operation of the programs will be uninterrupted or error-free.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period. If Tektronix is unable to provide a replacement that is free from defects in materials and workmanship within a reasonable time thereafter, Customer may terminate the license for this software product and return this software product and any associated materials for credit or refund.

THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THE PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESS OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPLACE DEFECTIVE MEDIA OR REFUND CUSTOMER'S PAYMENT IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

IMPORTANT

READ BEFORE OPERATING EQUIPMENT

This software is provided under license from Tektronix, Inc. Retention of this program for more than thirty (30) days or use of the program in any manner constitutes acceptance of the license terms.

CAREFULLY READ THE ENCLOSED SOFTWARE LICENSE AGREEMENT. If you cannot agree to the license terms, promptly contact the nearest Tektronix Field Office for return assistance.

TEKTRONIX SOFTWARE LICENSE AGREEMENT

THE PROGRAM, OR PROGRAMS, ENCODED OR INCORPORATED WITHIN EQUIPMENT, IS FURNISHED SUBJECT TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. RETENTION OF THE PROGRAM FOR MORE THAN THIRTY DAYS OR USE OF THE PROGRAM IN ANY MANNER WILL BE CONSIDERED ACCEPTANCE OF THE AGREEMENT TERMS. IF THESE TERMS ARE NOT ACCEPTABLE, THE UNUSED PROGRAM AND ANY ACCOMPANYING DOCUMENTATION SHOULD BE RETURNED PROMPTLY TO TEKTRONIX FOR A FULL REFUND OF THE LICENSE FEE PAID. (FOR INFORMATION REGARDING THE RETURN OF PROGRAMS ENCODED OR INCORPORATED WITHIN EQUIPMENT, CONTACT THE NEAREST TEKTRONIX SALES OFFICE.)

DEFINITIONS. "Tektronix" means Tektronix, Inc., an Oregon corporation, or local Tektronix’ legal entity that is supplying the equipment.

"Program" means the Tektronix software product (executable program and/or data) enclosed with this Agreement or included within the equipment with which this Agreement is packed.

"Customer" means the person or organization in whose name the Program was ordered.

LICENSE. Customer may:

a. Use the Program on a single machine at any one time;

b. If the Program is provided in connection with a floating-user license, the Program may be used on multiple machines provided that the user is authorized, and the total number of users at any one time does not exceed the total number of licensed concurrent users;

c. Modify the Program or merge it with another for use on the single machine; and

d. Copy the Program for archival or backup purposes, provided that no more than one (1) such copy is permitted to exist at any one time. If the Program is provided in connection with a floating-user license, the Program may be copied onto multiple machines for use by authorized users.

Each copy of the Program made by Customer must include a reproduction of any copyright notice or restrictive rights legend appearing in or on the copy of the Program as received from Tektronix.

Customer may not:

a. Use the Program on more than one machine at any one time, unless covered by a floating-user license or separate site license;

b. Transfer the Program to any person or organization outside of Customer or the corporation of which Customer is a part without the prior written consent of Tektronix, except in connection with the transfer of the equipment within which the programs are encoded or incorporated;

c. Export or re-export, directly or indirectly, the program, any associated documentation, or the direct product thereof, to any country to which such export or re-export is restricted by law or regulation of the United States or any foreign government having jurisdiction without the prior authorization, if required, of the Office of Export Administration, Department of Commerce, Washington, D.C. and the corresponding agency of such foreign government;

d. For object-code Programs only, reverse compile or disassemble the Program for any purpose; or

e. Copy the documentation accompanying the Program.

For Programs designed to reside on a single-machine and support one or more additional machines, either locally or remotely, without permitting the Program to be transferred to an additional machine for local execution, the additional machines shall be considered within the definition of "single machine". For programs permitting the Program to be transferred to an additional machine for local execution, a separate license shall be required for each such machine with which the Program may be used, or each concurrent user authorized under a floating-user license.

Title to the Program and all copies thereof, but not the media on which the Program or copies may reside, shall be and remain with Tektronix or others for whom Tektronix has obtained a respective licensing right.

Customer shall pay when due all property taxes that may now or hereafter be imposed, levied or assessed with respect to the possession or use of the Program or this license and shall file all reports required in connection with such taxes.

Any portion of the Program modified by Customer or merged with another program shall remain subject to these terms and conditions.

If the Program is acquired by or for an agency of the U.S. Government, the Program shall be considered computer software developed at private expense and the license granted herein shall be interpreted as granting Customer restricted rights in the Program and related documentation as defined in the applicable acquisition regulation.

THE PROGRAM MAY NOT BE USED, COPIED, MODIFIED, MERGED, OR TRANSFERRED TO ANOTHER EXCEPT AS EXPRESSLY PERMITTED BY THESE TERMS AND CONDITIONS.

UPON TRANSFER OF ANY COPY, MODIFICATION, OR MERGED PORTION OF THE PROGRAM, THE LICENSE GRANTED HEREIN IS AUTOMATICALLY TERMINATED.

TERM. The license granted herein is effective upon acceptance by Customer, and shall remain in effect until terminated as provided herein. The license may be terminated by Customer at any time upon written notice to Tektronix. The license may be terminated by Tektronix or any third party from whom Tektronix may have obtained a respective licensing right if Customer fails to comply with any term or condition and such failure is not remedied within thirty (30) days after notice hereof from Tektronix or such third party. Upon termination by either party, Customer shall return to Tektronix or destroy, the Program and all associated documentation, together with all copies in any form.

LIMITED WARRANTY. Tektronix warrants that the media on which the Program is furnished and the encoding of the Program on the media will be free from defects in materials and workmanship for a period of three (3) months from the date of shipment. If any such medium or encoding proves defective during the warranty period, Tektronix will provide a replacement in exchange for the defective medium. Except as to the media on which the Program is furnished, the Program is provided "as is" without warranty of any kind, either express or implied. Tektronix does not warrant that the functions contained in the Program will meet Customer's requirements or that the operation of the Program will be uninterrupted or error-free.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period. If Tektronix is unable to provide a replacement that is free from defects in materials and workmanship within a a reasonable time thereafter, Customer may terminate the license for the Program and return the Program and any associated materials for credit or refund.

THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THE PROGRAM IN LIEU OF ANY OTHER WARRANTIES, EXPRESS OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPLACE DEFECTIVE MEDIA, OR REFUND CUSTOMER'S PAYMENT IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY.

LIMITATION OF LIABILITY, IN NO EVENT SHALL TEKTRONIX OR OTHERS FROM WHOM TEKTRONIX HAS OBTAINED A LICENSING RIGHT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR CONNECTED WITH CUSTOMER'S POSSESSION OR USE OF THE PROGRAM, EVEN IF TEKTRONIX OR SUCH OTHERS HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

THIRD-PARTY DISCLAIMER. Except as expressly agreed otherwise, third parties from whom Tektronix may have obtained a licensing right do not warrant the program, do not assume any liability with respect to its use, and do not undertake to furnish any support or information relating thereto.

GENERAL. This Agreement contains the entire agreement between the parties with respect to the use, reproduction, and transfer of the Program.

Neither this Agreement nor the license granted herein is assignable or transferable by Customer without the prior written consent of Tektronix.

This Agreement and the license granted herein shall be governed by the laws of the state of Oregon.

All questions regarding this Agreement or the license granted herein should be directed to the nearest Tektronix Sales Office.

Table of Contents

Introduction

Applications

Software, Hardware, and User Prerequisites

Installation and Licensing

Compression Standards and File Types

How to Use the MTS4EA

Bitstream Syntax Debugging

Appendices

Glossary

Index

Preface

This manual describes the functions and use of the Tektronix MTS4EA Compressed Video Analyzer version 4.0. The manual is organized into the following sections:

- Section 1: Introduction
- Section 2: Applications
- Section 3: Software, Hardware and User Prerequisites
- Section 4: Installation and Licensing
- Section 5: Compression Standards and File Types
- Section 6: How to Use the MTS4EA
- Section 7: Bitstream Syntax Debugging
- Appendix A - Decoder Plugins for MTS4EA
- Appendix B - Tests of MTS4EA with MPEG-4 Normative and Donated Bitstreams
- Glossary
- Index

Related Material

Additional documentation, such as ReadMe files, may be included on the installation CD-ROM.

The following URLs access the Web sites for the standards organizations listed (the URLs listed were valid at the time of writing):

- MPEG-2 standards (International Organization for Standards)

  http://www.iso.ch/

- DVB standards (European Technical Standards Institute)

  http://www.etsi.org/

- ATSC standards (Advanced Television Systems Committee)

  http://www.atsc.org/

Product Naming Conventions

Although the order codes for this product are MTS4EAB for the Single local user license version and MTS4EAF for the Floating license version, the product is genearally referred to as the MTS4EA.

Manual Conventions

Naming conventions for the interface elements are based on standard Microsoft Windows naming conventions. Naming conventions for MPEG-2, ATSC, and DVB structures follow the conventions derived from the referenced standards documents.

Additionally:

- Mono-spaced text can indicate the following:
  - Text you enter from a keyboard
    Example: Enter the network identity (`http://TSMonitor01`)
  - Characters you press on your keyboard
    Example: Press `CTRL+C` to copy the selected text.
  - Paths to components on your hard drive
    Example: The program files are installed at the following location:
    `C:\Program Files\Tektronix\`

This manual is applicable to all versions of the MTS4EA including the MTS4EAB (Single local user license) and the MTS4EAF (Floating license).

Number Conventions

Within MTS4EA all numbers are decimal unless they are preceded by 0x, in which case they are hexadecimal.

For example:

- 16 [decimal] = 16 decimal
- 0x16 [hexadecimal] = 22 decimal

User Manual and Tutorials

The information in this manual is also provided in the MTS4EA Help system (press F1).

The PDF version of this manual can be also accessed from MTS4EA Help menu > User manual (PDF).

The PDF version of this manual is provided:

- Because it is sometimes easier/clearer to navigate through a PDF document than the standard Windows Help system
- So that you can easily print out additional copies of the manual

Tutorials are supplied as a separate PDF file accessible through the Help menu.

The PDF version of this manual and the tutorials are also on the installation CD-ROM.

Introduction

Introduction

Overview

MTS4EA Compressed Video ES Analyzer is a powerful real-time analytical tool for the investigation of compressed video and audio data that has been encoded using the H.264/AVC, MPEG-1/2, MPEG-4 (including AAC and HE-AAC), VC-1, AC-3, H.263+, and H.263 video standards.

MTS4EA operates as a standalone software application for use on a personal computer (PC) in a Windows environment. It can also be installed on the MTS400 MPEG Test Systems.

MTS4EA Compressed Video ES Analyzer is ordered using the following order codes:

- MTS4EAB (Single local user license)
- MTS4EAF (Floating license)

MTS4EA enables the display, analysis, debug, and optimization of the compression algorithm or equipment:

- Next generation (VC-1, H.264/AVC, MPEG-4 and 3GPP) and Legacy (MPEG-2, H.263 and H.263+) CODEC support
- Frame-by-frame and block-by-block analysis to allow easy CODEC comparison
- Easy-to-interpret detailed graphical displays (requires user-installed Microsoft Excel)
- Comprehensive semantic trace file output to determine block-by-block encoder decision making
- AV Delay measurement
- Audio decode and analysis
- Synchronized audio and video analysis
- Real-time and non real-time decoding and analysis of compressed video streams (dependent on PC performance)

- Elementary stream editing
- Batch mode to allow automated testing
- YUV decoded video output for baseband video analysis
- Extraction of elementary stream from transport stream
- Available as Single user local license for PC and Tektronix instruments or Server-based floating license

Applications

Applications

The key uses for a next generation compression ES analysis tool are as follows:

Equipment Manufacturers

- Semiconductor device designers and manufacturers
- Video and audio codec software and hardware developers
- STB, PVR, DVD consumer electronics developers for cable, satellite, terrestrial, and IP applications
- Video conferencing and communications equipment developers
- Mobile video and audio infrastructure and handset developers

Video Content Transmission and Distribution

- CODEC and equipment evaluation and comparison in cable, satellite, terrestrial, and IP applications
- Network operators
- Network equipment providers
- Application and service providers
- Streaming media applications
- Broadcasters checking for introduction of AV delay

Applications

- Analyze the performance (qualitative and quantitative) of audio and video compression algorithms, to assist with optimization, comparison, and evaluation.

- Check conformity and correctness of a compressed audio or video bit-stream against the various standards.

- Confirm interoperability between different vendor's audio and video compression algorithms.

- Provide comprehensive statistics that will enable the informed user to optimize audio and video compression algorithms.

- Display analysis results in text or graphical formats.

Software, Hardware, and User Prerequisites

Software, Hardware, and User Prerequisites

MTS4EA runs under the Microsoft Windows operating system and has been tested on:

- Windows XP (Home and Professional)
- Windows 2000 Professional

MTS4EA is not supported under any other operating system.

Audio Playout

In order to extract and play audio streams in MTS4EA the PC requires an audio device, correctly configured.

Personal Computer Prerequisites

MTS4EA will work satisfactorily on relatively modest PC systems (for example 800 MHz).

However, in order to decode and display CIF (Common Intermediate Format) video streams in real time the following minimum PC configuration is recommended for MPEG-4 and H.263 format video:

- Processor speed >1.5 GHz
- Memory at least 256 MB RAM
- Screen resolution at least 1024x768
- 400 MB minimum hard disk space

For H.264/AVC with CIF-size video (and larger) and MPEG-2 with D1-size video (and larger) the processing requirements are significantly higher and therefore a higher performance PC is required to display the video in real time: for these a PC with processor speed of at least 2.5 GHz is recommended.

When writing trace information to the hard disk, the limiting factor is the speed of the hard disk (regardless of the video standard), so the faster the better. Also, the video and Trace files can be large (hundreds of MB) so a large hard disk is needed:

- Hard disk speed: at least - 7200 rpm
- Hard disk size: as required for high-speed access to the video files to be analyzed plus Trace information

Trace File Writing/RAM disk

The real limitation on speed is when the trace information is being written to disk. This is the slowest operation; roughly 2 MB to 4 MB of data are written per video frame for the more extensive trace functions.

For most customers, this is not a problem, but if you want to have the highest speed when playing video and writing the trace files, a RAM disk can be used. This uses some of the system memory to emulate a disk drive.

There are a number of software products that offer this functionality; search on the Internet for ‘RAM disk’.

If a RAM disk is used, a memory configuration of at least 512 MB is recommended, with approximately 200 MB of RAM reserved for the RAM disk. This would allow a 200 MB Trace file to be saved on the RAM disk.

MTS400 MPEG Test Systems

You can install the MTS4EA on the Tektronix MTS400 family of MPEG test systems. The installation procedure is described in the Section 4.

User Prerequisites

The MTS4EA is sophisticated analysis software, which presents detailed information relating to the video and audio standards.

It therefore assumes that you understand these audio and video standards in significant detail and can therefore interpret the information presented by the MTS4EA.

A list of references are provided (see *Compression Standards and File Types*, Section 5).

MTS4EA Standalone Software Options

Table 3-1: Single License options

Item	Option	Description	
MTS4EAB		Base software with video standard package including: MPEG-4 Simple Profile, H.263+, H.263, TS extraction, single local user license. Includes English manual and CD.	
	PPD	Parallel Port software key (dongle)	Mandatory and mutually exclusive
	USB	USB software key (dongle)	
	LUD	Add MTS4EA to a current MTS400 MPEG Test system Transport Stream Compliance Analyzer software key, please provide serial number of your software key with your order.	
	M4SP	MPEG-4 Advanced Simple Profile (Levels 0 -5)	
	M2ML	MPEG-2 Main Profile Main Level and 4:2:2 profile at Main Level	
	M2HL	MPEG-2 Main Profile High Level, High Level 1440 and 4:2:2 profile at High Level	
	AVCE	H.264/AVC Baseline and Extended Profiles (Levels 1 – 5)	
	AVCM	H.264/AVC Main Profile (Levels 1 - 5)	
	AVCH	H.264/AVC High Profile with FRExt (10 bit, 4:2:2:, 4:4:4). Compatible with MTS4EA version 4.0 and above.	
	AVDM	AV Delay measurement Compatible with MTS4EA version 5.0 and above.	
	VC1	VC-1 (all Profiles, all Levels) and ASF extraction Compatible with MTS4EA version 4.0 and above.	
	AUD	Audio decode and analysis (incl. MPEG2 Layer 1 & 2, AAC, HE AAC) Compatible with MTS4EA version 5.0 and above.	
	SWS	First 12 months software subscription.	

Table 3-2: Floating License options

Item	Option	Description	
MTS4EAF		Base software with video standard package including: MPEG-4 Simple Profile, H.263+, H.263, TS extraction, floating license. Includes English manual and CD. This includes one license, for additional licenses please order multiple copies.	
	PPD	Parallel Port software key (dongle)	Mandatory and mutually exclusive
	FLT	Floating license token for addition to a mutli-license floating system at the time of order.	
	USB	USB software key (dongle)	
	M4SP	MPEG-4 Advanced Simple Profile (Levels 0 -5)	
	M2ML	MPEG-2 Main Profile Main Level and 4:2:2 profile at Main Level	
	M2HL	MPEG-2 Main Profile High Level, High Level 1440 and 4:2:2 profile at High Level	
	AVCE	H.264/AVC Baseline and Extended Profiles (Levels 1 - 5)	
	AVCM	H.264/AVC Main Profile (Levels 1 - 5)	
	AVCH	H.264/AVC High Profile with FRExt (10 bit, 4:2:2:, 4:4:4). Compatible with MTS4EA version 4.0 and above.	
	AVDM	AV Delay measurement Compatible with MTS4EA version 5.0 and above.	
	VC1	VC-1 (all Profiles, all Levels) and ASF extraction Compatible with MTS4EA version 4.0 and above.	
	AUD	Audio decode and analysis (incl. MPEG2 Layer 1 & 2, AAC, HE AAC) Compatible with MTS4EA version 5.0 and above.	
	SWS	First 12 months software subscription.	

Table 3-3: Stand alone options

Item	Opt	Description
MTS4EAUP		Upgrades for MTS4EA software including upgrade instructions, and software key (dongle) upgrade disk. See options for more details.
Upgrade options specific to standalone version		
MTS4EAUP	FLUP	Upgrade one current single local user license to floating. Current MTS4EA must be version 4.0 or above and have identical options. Not available for MTS4EA installed on an MTS400/430 or AD953A. If additional floating licenses are required then MTS4EAUP 4EAF
Upgrade and Support options specific to floating version		
MTS4EAUP	4EAF	Adds additional floating license to existing floating system. Base software with video standard package including: MPEG-4 Simple Profile, H.263+, H.263, TS extraction. Current MTS4EA must be version 4.0 or above and have identical options. Not Available
MTS4EAUP	SWS	First 12 months software subscription service when adding an additional floating license.
MTS4EAUP	SYNC	Synchronization of SWS option to new floating license SWS option, required when ordering new license on system which is under SWS agreement, where this SWS has run less than 8 months.
Upgrade and support option for both standalone and floating versions		
MTS4EAUP	SWS1	Software Subscription renewal
MTS4EAUP	UPG	Upgrade to latest version of MTS4EA base software and installed options. Includes CD and manual (Includes 1 year of software subscription.)
Upgrade options for adding new codecs to either standalone or floating versions		
MTS4EAUP	M4SP	MPEG-4 Advanced Simple Profile (Levels 0 -5)
MTS4EAUP	M2ML	MPEG-2 Main Profile Main Level and 4:2:2 profile at Main Level
MTS4EAUP	M2HL	MPEG-2 Main Profile High Level, High Level 1440 and 4:2:2 profile at High Level
MTS4EAUP	AVCE	H.264/AVC Baseline and Extended Profiles (Levels 1 - 5)
MTS4EAUP	AVCM	H.264/AVC Main Profile (Levels 1 - 5)
MTS4EAUP	AVCH	H.264/AVC High Profile with FRExt (10 bit, 4:2:2:, 4:4:4). Compatible with MTS4EA version 4.0 and above.
MTS4EAF	AVDM	AV Delay measurement Compatible with MTS4EA version 5.0 and above.
MTS4EAUP	VC1	VC-1 (all Profiles, all Levels) and ASF extraction Compatible with MTS4EA version 4.0 and above.
MTS4EAUP	AUD	Audio decode and analysis (incl. MPEG2 Layer 1 & 2, AAC, HE AAC) Compatible with MTS4EA version 5.0 and above.

Installation and Licensing

Installation and Licensing

This chapter describes MTS4EA license types and installation of the MTS4EA and license server.

License Types

MTS4EA licenses dictate the number of copies of MTS4EA and options that can be run at once. There are two different types of license: node-locked and floating.

In order to run the MTS4EA software you will need a software key with the correct license permissions.

NOTE. *The software key is very important as it contains the MTS4EA license and therefore embodies the value of MTS4EA. If the software key is lost, you will have to purchase another software key, which may be at full list price of the software.*

All options are visible in the installed MTS4EA menus. However, options not enabled by the license will be unavailable (grayed out). The status and scope of the license can be checked at any time by clicking on the Help menu option License manager….

Node Locked License

A node-locked license is locked to a particular PC, that is, MTS4EA will only run on the PC on which a software key is installed. When using node-locked licenses, each PC that is to run MTS4EA must have a separate node-locked license.

Software Key Based Floating License

The floating license uses a software key located on a license server to provide the license key.

Note the following system requirements:

- The software key will require a USB connection on the server machine.
- A license server process must be run on the server machine (this process has modest requirements for CPU load).
- Clients require TCP/IP access to the server (this could be provided via VPN between remote sites)

Floating License Installation

The installation process is license dependent. If the MTS4EA is to be used with a floating license, the server that is to host the license software must be set up first.

The floating license system enables a single software key (the DES DK2 dongle) to serve up licenses to multiple users connected over a network. The software key is preprogrammed with licenses before shipment.

To install the floating license system on the network, you must:

1. Install the DK2 Network Server software on a server (the license server).
2. On every client that is to use the floating license, install the MTS4EA software that also installs the DK2 client software.

Installing the License Server

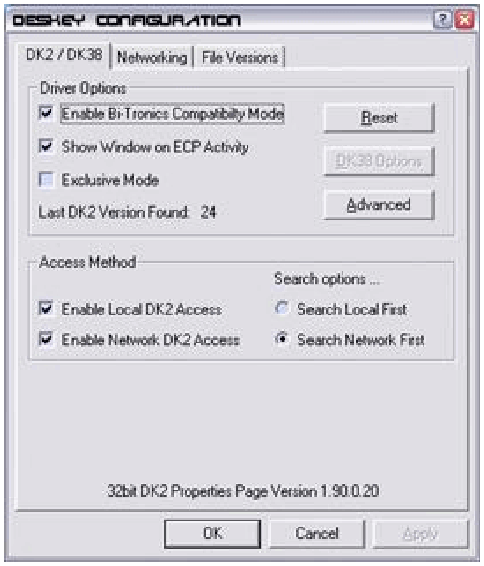
A license server hosts the software key, which contains one or more licenses. It is possible to have several license servers on the network, each with their own software key, forming a pool of licenses for the clients to use. It is also relatively easy to have a machine acting as both a server and a client. To allow a machine to act as a license server, the following procedure should be repeated on each machine:

3. Select a server machine and connect the software key (either via the USB port or parallel port depending on software key type).
4. On the server machine, insert the MTS4EA installation CD.

> ***NOTE**. When the CD is inserted, the MTS4EA setup screen will be displayed. Select the Close button at the bottom of the window to terminate the MTS4EA installation program.*

5. Using Windows Explorer, locate and open the `DES` directory. Run the program named `dnsrv32.exe`. During the installation process, accept the default entries.
6. Reboot the server machine if requested to do so at the end of the installation process.
7. Open the Control Panel (Windows Start > Settings > Control Panel) and select the DESkey icon. This will show the DESkey Configuration dialog, the first two tabs of which are shown in the following illustrations (note the version numbers shown near the bottom may vary).

***NOTE.** For the server installation, the DK2/DK38 tab and the top half of the Networking tab can be ignored; these refer to the client installation that shares the same configuration dialog.*



Driver Options

☑ Enable Bi-Tronics Compatibility Mode

☑ Show Window on ECP Activity

☐ Exclusive Mode

Last DK2 Version Found: 24

Access Method

☑ Enable Local DK2 Access

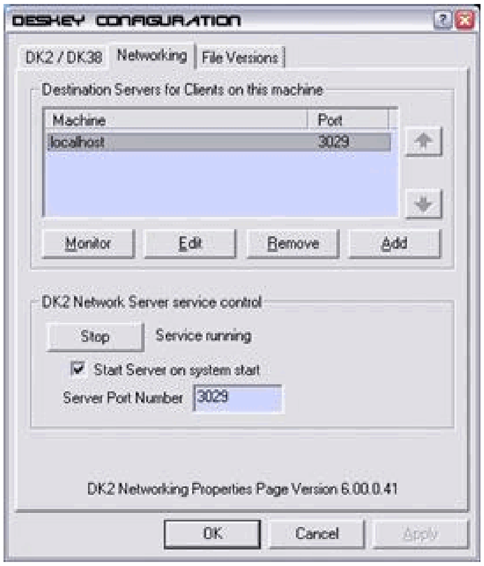
☑ Enable Network DK2 Access

Search options ...

○ Search Local First

● Search Network First

32bit DK2 Properties Page Version 1.90.0.20



Destination Servers for Clients on this machine

Machine	Port
localhost	3029

DK2 Network Server service control

Service running

☑ Start Server on system start

Server Port Number 3029

DK2 Networking Properties Page Version 6.00.0.41

8. Select the Networking tab. The field at the bottom labeled Server Port Number shows the port to be used for communication with any clients; the default is 3029. Press the OK button to finish.

However, depending on the policy of your IT department, port 3029 may be blocked by the network firewall, in which case you will need to consult your network administrator for another unrestricted port. Enter the new port and click the Apply button. You may then get a prompt asking about restarting the server; select Yes – this will stop and start the network server service using the new port. When the port has been applied, press the OK button to finish. This does not cause the server to reboot; it only restarts the license server.

Installing the MTS4EA Software

If the MTS4EA is to be used with a floating license, ensure that the following information is available:

- The network identity of the floating license server
- The server port number

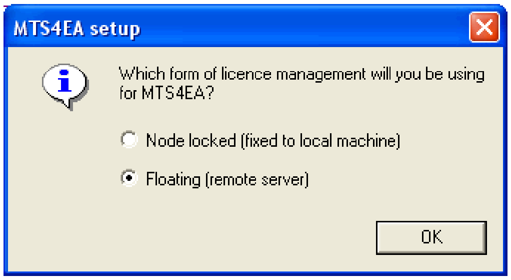
(These are set up in the *Floating License Installation* section.)

9. Insert the MTS4EA installation CD.

***NOTE**. If the installation process does not start automatically, run setup.exe from the root directory of the installation CD.*

10. At the bottom of the window showing the readme file, click the Install button to start the installation.

11. Select the required license management option.



Which form of licence management will you be using for MTS4EA?

○ Node locked (fixed to local machine)

● Floating (remote server)

a. Select OK to continue.

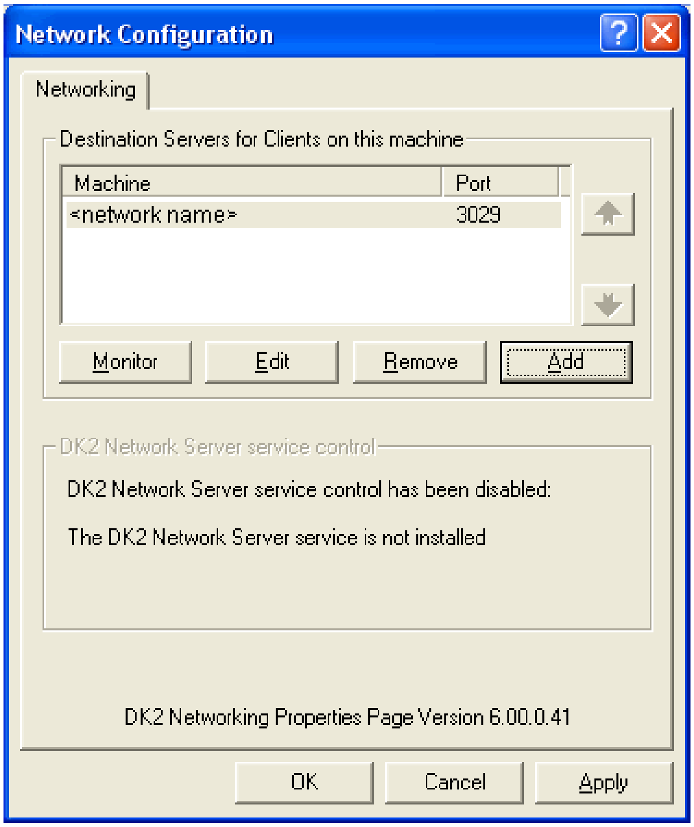
b. If Node locked is chosen, proceed to step 16.

c. If Floating is chosen, proceed to the next step.

12. An initial dialog box is shown. Press Next. The DK2 software key driver is installed.

***NOTE**. If you have installed the license server on this machine, launching the client installation may show a dialog box titled* DK2 Driver/Server Object *that asks if you want to start the DK2 Network Server - attach the software key and select* Yes.

13. At the end of the driver installation setup, a Network Configuration dialog box will appear (shown below).



Destination Servers for Clients on this machine

Machine	Port
<network name>	3029

DK2 Network Server service control

DK2 Network Server service control has been disabled:

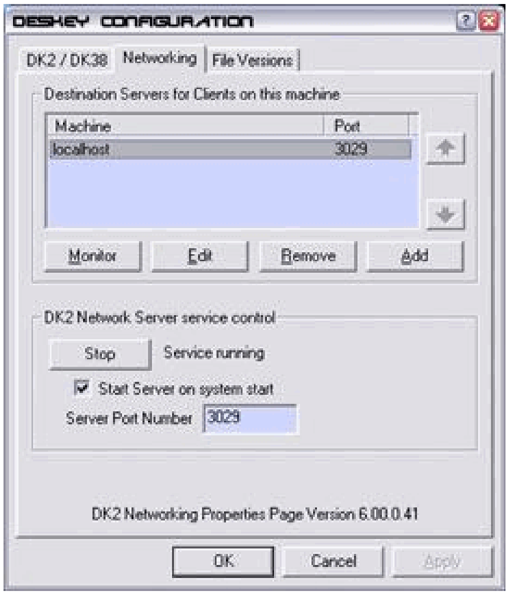
The DK2 Network Server service is not installed

DK2 Networking Properties Page Version 6.00.0.41

14. This enables the server(s) installed in the previous section to be added to the list of machines to query for licenses. If the required license server identity is not already listed, select Add and enter the required license server network identity and port number.

***NOTE**: If the DK2 installer detects a previous installation of the DK2 driver on this machine, the Network Configuration dialog box will not appear. In this case, allow the installation to proceed and complete. After installation, open the DESkey Networking Configuration dialog box (Windows Start > Settings > Control Panel > DESkey > Networking tab) and proceed as described below.*

If you are installing the MTS4EA client software on the license server, the server name to add is `localhost`. This is shown in the Networking Tab diagram of the DESkey Configuration dialog box in the License Server section above. The actual name of the machine can be used if preferred.



Machine	Port
localhost	3029

Service running

☑ Start Server on system start

Server Port Number 3029

DK2 Networking Properties Page Version 6.00.0.41

***NOTE.** Changes made to the Server list on the Networking tab will only take effect when the* Apply *button is pressed.*

15. Add all the servers that are to be accessible to this client, and then press the OK button to close the Network Configuration dialog box.

***CAUTION.** At this stage, depending on the history of the machine, a prompt may ask to reboot. Select* No*, otherwise the MTS4EA installation will not complete correctly and need to be rerun.*

16. The MTS4EA installation will continue. When finished, reboot the machine.

Testing the License Server

To ensure that the previous installations were successful, perform the following steps:

17. Remove any Tektronix DK2 software keys from the client machine and ensure the server machine has a suitably programmed software key connected.

18. Run the MTS4EA application using the desktop icon. If it starts, the license server is functioning correctly. If it does not start or displays “Invalid Product License” then check the following:

    On the server machine, ensure the software key is inserted correctly and the DK2 Network Server service is running. The latter can be checked by opening the DESkey Configuration dialog box from the Control Panel. The Networking tab shows the status of the service - if it is not running, then start it and try the MTS4EA application again.

Troubleshooting the License Server

If the service will not start: Reboot the machine and try again. If it still does not work, it may be a faulty or disabled port. For a USB software key, check the Device Manager for a DESkey entry. (Open the Windows Control Panel, select the System icon, on the Hardware tab click the Device Manager button. Any problem will show up as a yellow triangle next to the DESkey entry.) For a parallel port software key, check that the port is enabled in the machine's BIOS settings.Try a different port connection if more than one is available.

It could also be a faulty software key, try connecting the software key to the client machine and run the MT4EA software again (ensure the Enable Local DK2 Access checkbox is selected in the DK2/DK38 tab of the DESkey configuration dialog box). If this works, the software key is working, so try another server machine. If this does not work, contact Tektronix Customer Support.

If the service will start but the mts4ea will not run: Check with the network administrator that the port being used is not being blocked by a firewall. If it is either get it unblocked or choose another port, changing it in both the server and all clients; remember to press Apply in the associated dialog boxes before closing them.

On the client machine, check the machine name entered in the client server list is present and correct. If you entered it manually, try navigating to it using the Browse feature to confirm that it is correct.

Check that the port number specified by the server is the same as the port number specified in the client’s server list.

On the client machine, ensure the Enable Network DK2 Access checkbox is ticked on the DK2/DK38 tab of the DESkey Configuration dialog (see page 4–3).

Compression Standards and File Types

Compression Standards and File Types

This chapter gives:

- A description of the standards (and elements within those standards), that MTS4EA supports, both video and audio
- The file types MTS4EA can work with, both video and audio
- The exact titles and numbers of the standards on which the MTS4EA is based

Supported Video Compression Standards

The video compression standards supported in this version of the MTS4EA are:

- H.264/AVC
- VC-1
- MPEG-4 Part 2
- MPEG-2 Part 2
- H.263+, H.263

The MTS4EA will open container files (containing the above types of video):

- MPEG-2 Transport Stream (H.264/AVC, VC-1 Advanced profile, MPEG-2 Program Stream, MPEG-2 Packetized Elementary Stream, MPEG-2 Elementary Stream only)
- MP4 (MPEG-4 Part 2, H.263 Baseline only and H.264)
- 3GPP (MPEG-4 Part 2, H.263 Baseline only and H.264)
- VOB/MPEG-2 Program Stream (MPEG-2 only)
- Microsoft ASF files
- RCV files

The MTS4EA will also open uncompressed video files: see *Opening an Uncompressed Video File (any File Extension)* in the *How to Use the MTS4EA* section for more information.

The MTS4EA will also open audio files of the standards given in *Supported Audio Compression Standards* on page 5-9, and container files that have these types of audio within them.

NOTE. *Not all container files support all types of audio listed under* Supported Audio Compression Standards *on page 5-9, see also* Permitted Audio Types/Format *on page 5-19.*

H.264/AVC

This version of MTS4EA supports the following elements of the H.264/AVC standard (see Reference [13] under *Standards References* on page 5-21).

H.264/AVC Profiles, Levels. The following Profiles and Levels are supported:

- Baseline Profile
  - All Levels from 1 to 5.1 inclusive
- Extended Profile
  - All Levels from 1 to 5.1 inclusive
- Main Profile
  - All Levels from 1 to 5.1 inclusive
- High Profile
  - All Levels from 1 to 5.1 inclusive
- High 10-bit Profile
  - All Levels from 1 to 5.1 inclusive
- High 4:2:2 Profile
  - All Levels from 1 to 5.1 inclusive
- High 4:4:4 Profile
  - All Levels from 1 to 5.1 inclusive

H.264/AVC Tools. (See Reference [13] under *Standards References* on page 5-21 for a description of these tools.)

All tools are supported, as permitted in the H.264/AVC standard, listed below.

Tool	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4
Profile and level indications:							
profile_idc	66	88	77	100	110	122	144
constraint_set0_flag	1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
constraint_set1_flag	0 or 1	0 or 1	1	0 or 1	1	1	1
constraint_set2_flag	0 or 1	1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1
constraint_set3_flag	0 or 1	0 or 1	0 or 1	0	0	0	0
level_idc	10 - 51	10 - 51	10 - 51	10 - 51	10 - 51	10 - 51	10 - 51
Slice types:							
I	Yes	Yes	Yes	Yes	Yes	Yes	Yes
P	Yes	Yes	Yes	Yes	Yes	Yes	Yes
B	No	Yes	Yes	Yes	Yes	Yes	Yes
SP	No	Yes	No	No	No	No	No
SI	No	Yes	No	No	No	No	No
NAL unit types:	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1 - coded slice of a non-IDR picture	No	Yes	No	No	No	No	No
2 - coded slice data partition A	No	Yes	No	No	No	No	No
3 - coded slice data partition B	No	Yes	No	No	No	No	No
4 - coded slice data partition C	Yes	Yes	Yes	Yes	Yes	Yes	Yes
5 - coded slice of an IDR picture	Yes	Yes	Yes	Yes	Yes	Yes	Yes
6 - supplemental enhancement information	Yes	Yes	Yes	Yes	Yes	Yes	Yes
7 - sequence parameter set	Yes	Yes	Yes	Yes	Yes	Yes	Yes
8 - picture parameter set	Yes	Yes	Yes	Yes	Yes	Yes	Yes
9 - access unit delimiter	Yes	Yes	Yes	Yes	Yes	Yes	Yes
10 - end of sequence	Yes	Yes	Yes	Yes	Yes	Yes	Yes
12 - end of stream	Yes	Yes	Yes	Yes	Yes	Yes	Yes
13 - filler data	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Structural:							
data partitioning (NALU type 2-4)	No	Yes	No	No	No	No	No
interlace (frame_mbs_only_flag)	No	Yes (L2.1 - 4.1)	Yes (L2.1 - 4.1)	Yes (L1-2 & 4.2 - 5.1)	Yes (L1-2 & 4.2 - 5.1)	Yes (L1-2 & 4.2 - 5.1)	Yes (L1-2 & 4.2 - 5.1)
arbitrary slice order	Yes	Yes	No	No	No	No	No
slice groups (num_slice_groups_minus1)	1 - 8	1 - 8	1	1	1	1	1
redundant coded pictures (redundant_pic_cnt_present_flag)	Yes	Yes	No	No	No	No	No
Weighted prediction:							
weighted_pred_flag	0	=> 0	=> 0	=> 0	=> 0	=> 0	=> 0
weighted_pred_idc	0	=> 0	=> 0	=> 0	=> 0	=> 0	=> 0
Entropy coding (entropy_coding_mode_flag):							
CAVLC							
CABAC	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	No	No	Yes	Yes	Yes	Yes	Yes
B frames:							
direct_8x8_inference_flag	n/a	1	0 (L1-2.2) 1 (L3-5.1)	1 (L3– 5.1)	1 (L3–5.1)	1 (L3–5.1)	1 (L3–5.1)
MinLumaBiPredSize	n/a	8x8 L3.1 - 5.1	8x8 L3.1 - 5.1	8x8 L3.1 - 5.1	8x8 L3.1 - 5.1	8x8 L3.1 - 5.1	8x8 L3.1 - 5.1

Tool	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4
Fidelity range:							
chroma format (chroma_format_idc)	0	0	0	0 - 1	0 - 1	0 - 1	0 - 3
bit depth (bit_depth_luma_minus8, bit_depth_chroma_minus8)	0	0	0	0	0 - 2	0 - 2	0 - 4
transform bypass (qpprime_y_zero_transform_bypass_flag)	0	0	0	0	0	0	0 or 1

No = not allowed; n/a = not applicable

VC-1

This version of MTS4EA supports the following elements of the VC-1 standard (see Reference [17] under *Standards References* on page 5-21).

VC-1 Profiles, Levels. The following Profiles and Levels are supported (this is all Profiles and all Levels):

- Simple Profile:
  - Low
  - Medium
- Main Profile:
  - Low
  - Medium
  - High
- Advanced Profile:
  - L0
  - L1
  - L2
  - L3
  - L4

MPEG-4

This version of MTS4EA supports the following elements of the MPEG-4 standard (see Reference [1] under *Standards References* on page 5-21).

***NOTE.** The MTS4EA support for MPEG-4 has been updated to the 2004 version of the MPEG-4 standard, which has a number of differences from the previous version.*

MPEG-4 Profiles, Levels. The following Profiles and Levels are supported:

- Simple Profile:
  - Level 0 (see note below)
  - Level 1
  - Level 2
  - Level 3
- Advanced Simple Profile:
  - Level 0
  - Level 1
  - Level 2
  - Level 3
  - Level 3b (see Reference [6], *Standards References* on page 5-21)
  - Level 4
  - Level 5
- Main Profile:
  - Level 2

***NOTE.** Level 0, which is not in the MPEG-4 Standard Reference [1] (see* Standards References *on page 5-21), is an addition to Simple Profile. Level 0 is targeted at mobile applications: for example, with a maximum picture size of QCIF and maximum frame rate of 15 frames per second.*

MPEG-4 tools. (See the following standard for a description of the MPEG-4 tools: MPEG-4 Part 2 (Visual), Reference [1] Table 9-1.)

For MPEG-4 Simple Profile, Advanced Simple Profile and Main Profile, the following tools are supported:

Tool	Simple Profile	Advanced Simple Profile	Main Profile
Basic: ▪ I-VOP ▪ P-VOP ▪ AC/DC prediction ▪ 4 Motion Vectors ▪ Unrestricted Motion Vectors	Y	Y	Y
Short header	Y	Y	Y
Error resilience: ▪ Slice Resynchronization ▪ Data Partitioning ▪ Reversible VLC	Y	Y	Y
B-VOPs	▨	Y	N
Method 1/Method 2 quantization	▨	Y	Y
Global Motion Compensation: ▪ Mode 0 ▪ Mode 1 ▪ Mode 2 ▪ Mode 3	▨	Y	▨
Quarter Sample Motion Compensation (quarter pel)	▨	Y	▨
Interlace	▨	Y (L4, 5 only)	N

▨ Means that this tool is not allowed in the profile concerned.

NOTE. *MPEG-4 Simple Profile Short Header is decoded by MTS4EA as H.263 Baseline, as defined by the MPEG-4.*

The following additional tools are not supported for Main Profile:

- P-VOP based temporal scalability:
  - Rectangular
  - Arbitrary shape
- Binary shape
- Gray shape
- Sprite

MPEG-2

This version of MTS4EA supports the following elements of the MPEG-2 standard (see Reference [14] under *Standards References* on page 5-21).

MPEG-2 Profiles, Levels. The following Profiles and Levels are supported, subject to the restrictions given under *Permitted Audio Types/Format* on page 5-19.

- Main Profile:
  - Main Level
  - High Level
  - High Level 1440
- 4.2.2 Profile:
  - Main Level
  - High Level

MPEG-2 tools. For MPEG-2 Main Profile at Main Level, at High Level and at High Level 1440, the following tools are supported:

Tool	Main Level	High Level	High Level 1440
I-frames	Y	Y	Y
P-frames	Y	Y	Y
B-frames	Y	Y	Y
Field-coded pictures (Interlaced)	Y	Y	Y

Tool	Main Level	High Level	High Level 1440
Frame-coded pictures: With field or frame order MacroBlocks	Y	Y	Y
Layers: GOP Picture Slice MacroBlock Block	Y	Y	Y

***NOTE.** Higher syntactic structures are also supported within MTS4EA - PES and VOB - see* MPEG-2 Higher Syntactic Structures *on page 5-13.*

H.263+, H.263

The following H.263 standards are supported:

- H.263 baseline standard (Reference [2] under *Standards References* on page 5-21).
- H.263+ (see Reference [2] under *Standards References* on page 5-21), including Annexes:
  - A: Inverse transform accuracy specification
  - B: Hypothetical reference decoder
  - C: Considerations for Multipoint
  - D: Unrestricted Motion Vector mode
  - F: Advanced Prediction mode
  - I: Advanced Intra Coding mode
  - J: Deblocking filter mode
  - K: Slice Structured mode
  - S: Alternative Inter VLC mode
  - T: Modified Quantization mode

No other annexes are supported.

Supported Audio Compression Standards

MTS4EA supports audio extraction and playout. MTS4EA also supports audio stream analysis functionality for all supported standards except Dolby Digital (AC-3). Supported standards are:

- MPEG-1 Part 3 audio
  (References [23] in *Standards References* on page 5-21)

- MPEG-2 Part 3 audio
  (References [19] in *Standards References* on page 5-21)

- MPEG-2 Part 7 AAC
  (Reference [20] in *Standards References* on page 5-21)

- MPEG-4 Part 3 audio
  (Reference [21] in *Standards References* on page 5-21)

- Dolby Digital (AC-3)
  (Reference [25] in *Standards References* on page 5-21)

MTS4EA also supports audio files in the following container formats:

- MPEG-4 Part 3 ADTS files
  (Reference [22] in *Standards References* on page 5-21)

- MPEG-4 Parts 1 and 14
  (References [11] and [24] in *Standards References* on page 5-21)

- MPEG-2 Transport, Program, PES streams
  (Reference [15] in *Standards References* on page 5-21)

- 3GPP (Reference [12] in *Standards References* on page 5-21)

MPEG-1 Part 3 audio

MTS4EA supports the MPEG-1 Part 3 audio standard, as defined in Reference [23] in *Standards References* on page 5-21. MTS4EA supports the following layers of this standard:

- Layer I

- Layer II

MTS4EA supports monophonic and stereophonic streams in this standard.

MPEG-2 Part 3 audio

MTS4EA supports the MPEG-2 Part 3 audio standard, as defined in Reference [19], *Standards References* on page 5-21. MTS4EA supports the extension of MPEG-1 part 3 audio to LSF (Low Sampling Frequency) for the following layers:

- Layer I
- Layer II

MTS4EA supports monophonic and stereophonic streams in this standard.

MPEG-2 Part 7 AAC audio

MTS4EA supports the MPEG-2 Part 7 AAC audio standard, as defined in Reference [20], *Standards References* on page 5-21. MTS4EA supports the following profiles of this standard:

- Main
- LC (Low Complexity)
- LTP (Long Term Prediction)

MTS4EA supports monophonic and stereophonic streams in this standard.

MPEG-4 Part 3 audio

MTS4EA supports the MPEG-4 Part 3 audio standard, as defined in Reference [21], *Standards References* on page 5-21. MTS4EA supports the following profiles of this standard:

- Main
- LC (Low Complexity)
- LTP (Long Term Prediction)
- HE-AAC (High Efficiency, ‘AAC plus’), low complexity profile with SBR (Spectral Band Replication)

MTS4EA supports monophonic and stereophonic streams in this standard.

MPEG-4 Part 3 ADTS files

MTS4EA supports the Audio Data Transport Stream (ADTS) for AAC audio, as defined in Reference [22], *Standards References* on page 5-21.

Permitted Video Types/Formats

MTS4EA analyzes and displays many file types, such as:

- Video files containing video Elementary Streams
- Container files, such as ASF files, RCV files, MP4/3GPP files, MPEG-2 Transport Streams, and MPEG-2 Program streams, which include video, audio and other data
- MTS4EA Trace files (which have a .vpt file extension)
- Uncompressed video files, such as YUV, RGB or grayscale color models, 8-16 bit sample depth, various chroma sub-sampling formats
- Any data file, using MTS4EA HexView (hexadecimal file viewer, on the Analysis menu)

The primary analysis functions of MTS4EA are performed on video Elementary Streams, and features are provided to extract these from within container files such as MP4 and 3GPP files.

Microsoft ASF Files

ASF files typically contain video, audio and other data. An ASF file can contain more than one video stream. See Reference [18] in *Standards References* on page 5-21.

RCV Files

RCV files contain video in the VC-1 format. See Reference [17] in *Standards References* on page 5-21

MP4 Files

MP4 files typically contain video, audio and other data. An MP4 file can contain more than one video stream; these streams can be located consecutively or split up in many areas of the MP4 file.

The MP4 files that the MTS4EA analyzes should conform to the relevant part of the MPEG-4 standard – Reference [11] in *Standards References* on page 5-21.

By default, the MTS4EA looks for MP4 files with an .mp4 file extension, although any file name can be used: MTS4EA will determine that it is an MP4 file by looking through the file contents.

The video streams can be extracted from the MP4 file and analyzed directly, or extracted and analyzed later. For information on opening MP4 files, extracting and saving the video streams (see *Open Stream* in the *How to Use the MTS4EA* section). The structure of the MP4 file can be examined, using the View file structure... option in the Analysis menu (see *View file structure* in the *How to Use the MTS4EA* section for more information).

3GPP Files

3GPP files are a subset of MPEG-4 files, with an indicator in the file to denote that it is a 3GPP file.

3GPP files should conform to the standard concerned – Reference [12] in *Standards References* on page 5-21.

By default, the MTS4EA looks for 3GPP files with a .3gp file extension, although any file name can be used: MTS4EA will determine that it is a 3GPP file by looking through the file contents.

The video streams are extracted from 3GPP files and analyzed directly or extracted and analyzed later in the same way as MP4 files. For information on opening 3GPP files and extracting and saving the video streams, see *Open Stream* in the *How to Use the MTS4EA* section.

The structure of the 3GPP file can be examined, using the View file structure... option in the Analysis menu (see *View file structure* in the *How to Use the MTS4EA* section for more information).

MPEG-2 Transport Stream Files

MPEG-2 Transport Streams typically contain video, audio, and other data. An MPEG-2 Transport Stream can contain more than one video stream - in fact hundreds or more; these streams can be located largely consecutively in the Transport Stream packets or split up in many areas of the Transport Stream.

Transport Streams with 188 bytes per packet, 196 bytes per packet, 204 bytes per packet and 208 bytes per packet can be analyzed.

MPEG-2 Higher Syntactic Structures

MPEG-2 Program Stream and PES files. An MPEG-2 Program Stream contains one or more Packetized Elementary Streams (PES), with a common time base. (The Program Stream is designed for use in a relatively error-free environment.)

The format of the Program Stream is defined in Part 1 of the MPEG-2 standard, Reference [15] in *Standards References* on page 5-21.

A Program Stream can contain a complete Elementary Stream, or it can contain only part of an Elementary Stream.

MTS4EA will open MPEG-2 Program Stream files (typically with a file extension of .mpg) and display the structure of the Program Stream, such as the PES packets inside.

In accordance with the MPEG-2 standard (see Reference [15] in *Standards References* on page 5-21), the Program Stream file will start with a 32-bit start code: 0x000001ba, 0000 0000 0000 0000 0000 0001 1011 1010.

MPEG-2 VOB files (DVD). VOB files are used on a DVD to store the video and audio; a VOB contains a Program Stream.

A single video Elementary Stream (for example, a movie) is usually distributed among a number of VOB files.

NOTE. *You can specify the different VOBs that contain a single Elementary Stream, so that the MTS4EA can extract the Elementary Stream from the different VOBs.*

MTS4EA will open MPEG-2 VOB files (typically with a file extension of .vob) and display the structure of the Program Stream, such as the PES packets inside.

See Reference [16] in *Standards References* on page 5-21.

NOTE. *When opening a VOB, there is no requirement that the VOB begins with an I-frame. In this situation, the video will not display correctly until an I-frame is reached.*

Encrypted (scrambled) VOBs cannot be opened.

Uncompressed Video File Format

The general uncompressed video file format is as follows:

- No headers of any kind (no file or frame headers)
- Concatenated planar image data
- Row raster order (top picture row first)
- Unsigned samples

For 8-bit sample depth:

- One byte per sample

For 9-16 bit sample depth:

- Two bytes per sample
- Both little- and big-endian byte orders supported

For YUV format:

- Concatenated Y, U and V planes
- U and V planes sub-sampled as required
- Y plane samples are unsigned
- U and V plane samples are unsigned with a DC offset of 2^{n-1}, where *n* is the chroma sample bit depth

For RGB format:

- Concatenated R, G and B planes

For grayscale format:

- Luma plane only

Compressed Video File Format

The formats of the raw compressed video data files (for example called Elementary Streams, typically with a file extension of .m4v or .cmp in MPEG-4) that MTS4EA analyzes are defined in the relevant standards. Proprietary file formats are not supported.

This means that the compressed video file to be analyzed should not be an AVI file (which puts a wrapper around the encoded video and audio).

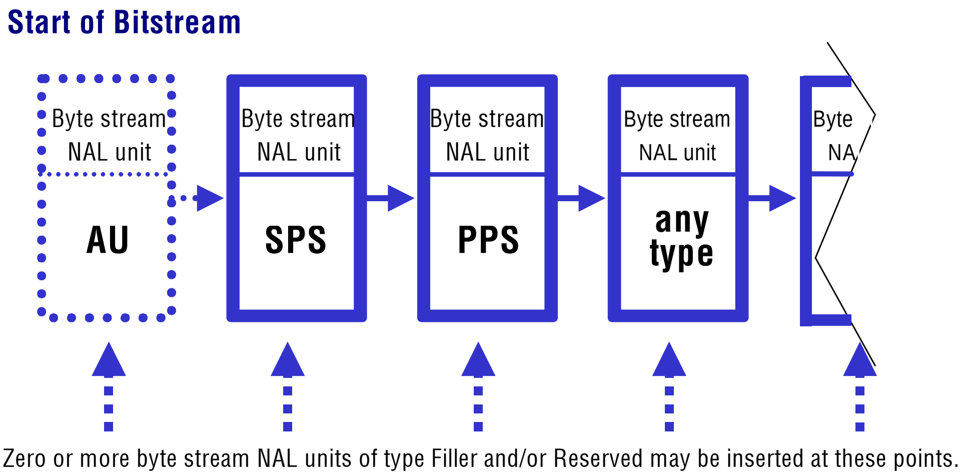
The compressed video elementary stream file cannot contain audio, although the original container file (such as MP4 or 3GPP) may have contained audio.

Once the video is extracted, the MTS4EA will automatically determine the format of the compressed video in the file by searching through the video and looking for the appropriate start codes. Accordingly, the file name does require a particular file extension to be decoded correctly.

H.264/AVC Byte Stream file format. In accordance with the H.264/AVC standard (see Reference [13] in *Standards References* on page 5-21), the video file must start as given in the following diagram.

The abbreviations mean:

AU	access_unit_delimiter_rbsp_nal_unit
SPS	sequence_parameter_set_nal_unit
PPS	picture_parameter_set_nal_unit



For example, if the bitstream does not contain the optional items at the start then the first six bytes in an Extended profile byte stream file will be:
00 00 00 01 67 58 (all values in hexadecimal)

VC-1 Advanced Profile and Elementary Stream file format. Within the VC-1 standard (see Reference [18] in *Standards References* on page 5-21), only the Advanced Profile can exist as a separate Elementary Stream. (The Simple and Main Profiles are required to be part of a container file - such as RCV, ASF or MPEG-2 TS - so that the required header information is stored.)

MTS4EA fully supports the VC-1 implementations of the Simple, Main and Advanced Profiles in each standard.

(See the standards in Reference [17] in *Standards References* on page 5-21 for details of the start codes and syntax permissible for each of these.)

MPEG-4 Elementary Stream file format. In accordance with the MPEG-4 standard (see Reference [1] in *Standards References* on page 5-21), the file will start with a valid header start code that contains configuration information such as:

	`visual_object_sequence_start_code`	(0xB0)
or	`visual_object_start_code`	(0xB5)
or	`video_object_layer_start_code`	(0x20 - 0x2F)
or	`video_object_start_code`	(0x00 - 0x1F)

For MPEG-4 Short Header video there is no header code other than the 22-bit start code:

- `video_plane_with_short_header()` (see below)

The start codes actually used will vary from one stream to another.

These start codes (except Short Header) may be followed by further configuration data until the start code for the video data is reached, which must start with one of the following functions:

- `Group_of_VideoObjectPlane()` start code: 0xB3
- `VideoObjectPlane()` start code: 0xB6

See the MPEG-4 standard Reference [1] (in section 6.1.2) in *Standards References* on page 5-21 for more information.

Note the entry points `MeshObject()` and `fba_object()` also given in section 6.1.2 of the MPEG-4 standard are not supported, but also these are non-allowable tools in the Profiles and Levels supported by MTS4EA.

This means that for

- `Group_of_VideoObjectPlane()`
- `VideoObjectPlane ()`

the 24 bits in the video file before the start code must be the Start Code Prefix, : 0000 0000 0000 0000 0000 0001. This is then followed by the 8-bit start code value.

However, for

- `video_plane_with_short_header()`

the first 22 bits in the video stream are the same as for H.263+ and H.263 (see below).

MPEG-2 Elementary Stream file format. In accordance with the MPEG-2 standard (see Reference [15] in *Standards References* on page 5-21), the Elementary Stream file will start with a 32-bit start code:
Hex: 0x000001b3; Bin: 0000 0000 0000 0000 0000 0001 1011 0011.

H.263+, H.263 file format. The first bits in an H.263+/H.263 compressed file should be the 22-bit PSC (Picture Start Code): 0000 0000 0000 0000 1000 00.

Source Video Format

The above means that when the video was encoded, the uncompressed source format of the video file to be encoded should have been YUV 4:2:0 - that is, with 4 bytes of Y data (luminance) for every 1 byte of U and 1 byte of V data (color differences). (The color difference data is sub-sampled by 2 in horizontal and vertical directions.)

For H.264/AVC High Profiles, the following alternative image formats are also supported:

- YUV, RGB or grayscale color models
- Alternative YUV chroma sub-sampling: 4:2:0, 4:2:2, or 4:4:4
- Alternative image sample bit depths: from 8 to 12 bits

It is up to the encoder to take the input data and order it correctly, so that in the compressed bitstream, the pixel data must appear with the top left pixel first, followed by the pixel to the right. If the video data stream input to the encoder does not follow this order then it is up to the encoder to re-arrange the data order.

Video Image Size

The maximum size of the video images that can be analyzed within MTS4EA is effectively up to 16384x16384 pixels.

For each standard, there is a maximum size and within this limit, MTS4EA will analyze video to the maximum size allowed in the video standard concerned, for example:

- 4096 pixels wide x 2304 pixels high in H.264/AVC
- 1920 pixels wide x 1088 pixels high in MPEG-2 Main Profile @ High Level

There is no minimum size.

Video File Size

The maximum video file size that can be analyzed is effectively limited only by the disk space on your computer (the limit to the size is actually 1 million Terabytes [2^{60}] or 1 billion Gigabytes).

MTS4EA will open the first part of the video file and start playing and analyzing this immediately if possible: in some cases, MTS4EA must search through a certain amount of the file before there is sufficient data for display.

In addition, the results from analysis of the input video file are cached in MTS4EA memory in the Step-back buffer. If the results for the section of video to be displayed/analyzed are not in the cache then this is indicated on-screen and MTS4EA will then pause while loading the relevant data in the cache.

PAL/NTSC

Both PAL and NTSC format video can be analyzed with MTS4EA.

Permitted Audio Types/Format

Audio is supported under the following container formats, as described in the previous section:

- MPEG-4 Part 3 ADTS files
- MPEG-2 Transport Stream, Program Stream, Packetized Elementary Stream (PES)
- 3GPP
- MP4

MPEG-2 Transport Stream files

MPEG-2 Transport Streams typically contain video, audio and other data. An MPEG-2 Transport Stream can contain more than one media stream - in fact hundreds or more; these streams can be located largely consecutively in the Transport Stream packets or split up in many areas of the Transport Stream.

Transport Streams with 188 bytes per packet, 196 bytes per packet, 204 bytes per packet and 208 bytes per packet can be analyzed.

MPEG-2 Program Stream and PES files

An MPEG-2 Program Stream contains one or more Packetised Elementary Streams (PES), with a common time base. (The Program Stream is designed for use in a relatively error-free environment.)

The format of the Program Stream is defined in Part 1 of the MPEG-2 standard (see Reference [15] in *Standards References* on page 5-21).

A Program Stream can contain a complete Elementary Stream, or it can contain only part of an Elementary Stream.

The MTS4EA will open MPEG-2 Program Stream files (typically with a file extension of .mpg) and display the structure of the Program Stream, such as the PES packets inside.

In accordance with the MPEG-2 standard (see Reference [15] in *Standards References* on page 5-21), the Program Stream file will start with a 32-bit start code: 0x000001BA, for example, 0000 0000 0000 0000 0000 0001 1011 1010

3GPP files

3GPP files are a subset of MPEG-4 files, with an indicator in each 3GPP file to denote that it is a 3GPP file.

3GPP files should conform to the standard concerned (see Reference [12] in *Standards References* on page 5-21).

By default, the MTS4EA looks for 3GPP files with a .3gp file extension, although any file name can be used; the MTS4EA will determine that it is a 3GPP file by looking through the file contents.

The audio streams are extracted from 3GPP files and analyzed directly or extracted and analyzed later in the same way as for MP4 files. For information on opening 3GPP files, extracting and saving the audio streams see the *How to Use the MTS4EA* section.

The structure of the 3GPP file can be examined, using the View file structure... option on the Analysis menu: see the *How to Use the MTS4EA* section for more information.

MP4 files

MP4 files typically contain video, audio and other data. An MP4 file can contain more than one video stream; these streams can be located consecutively or split up in many areas of the MP4 file.

The MP4 files that the MTS4EA analyzes should conform to the relevant part of the MPEG-4 standard
(see Reference [11] in *Standards References* on page 5-21).

By default, the MTS4EA looks for MP4 files with an .mp4 file extension, although any file name can be used; the MTS4EA will determine that it is an MP4 file by looking through the file contents.

The audio streams can be extracted from the MP4 file and analyzed directly, or extracted and analyzed later. For information on opening MP4 files and extracting and saving the audio streams see the *How to Use the MTS4EA* section.

The structure of an MP4 file can be examined, using the View file structure... option on the Analysis menu; see the *How to Use the MTS4EA* section for more information.

Standards References

[1] MPEG-4 Part 2 (Visual): standard number ISO/IEC 14496-2:2004; ISO title: Information technology - Coding of audio-visual objects: Part 2: Visual, 3rd Edition 2004-06-01; plus 14496-2:2004 Technical Corrigendum 1 Published 2004-06-15

[2] H.263: Video Coding for Low Bit Rate Communication. International Telecommunication Union (ITU) 1998

[3] H.261: Video Codec for AudioVisual Services at px64 kbit/s. International Telecommunication Union (ITU) 1994

[4] MPEG-4 Video Verification Model version 18.0: document reference number ISO/IEC JTC1/SC29/WG11 N3908 dated January 2001

[5] MPEG-4 Part 2 (Visual) ISO/IEC 14496-2 Amendment 2, 2002-02-01: Streaming video profile (contains Advanced Simple Profile)

[6] MPEG-4 Part 2 (Visual) ISO/IEC 14496-2:2001 Final Draft Amendment 3 FDAM 3:2003(E): New levels and tools for MPEG-4 visual (contains Advanced Simple Profile Level 3b)

[7] MPEG-4 Part 2 (Visual) ISO/IEC 14496-2 Microsoft reference software: FDAM1-2.3-001213 version 2 dated July 3rd 2000

[8] MPEG-4 Part 4 (Conformance Testing) ISO/IEC Study of CD 14496-4 N3067 1999-03-18, Visual clause w3067_4(v)

[9] MPEG-4 Part 4 (Conformance Testing) ISO/IEC 14496-4 MPEG-4 Normative ISO bitstreams dated 05/11/2001, specified in sections 4.5.3.1 and 4.5.7 of Reference [8]

[10] MPEG-4 Part 4 (Conformance Testing) ISO/IEC 14496-4 MPEG-4 Donated bitstreams dated 14/07/2000, referred to in section 4.5.8 of Reference [8]

[11] MPEG-4 Part 1 (Systems) ISO/IEC 14496-1: Information technology - Coding of audio-visual objects: Part 1: Systems, 3rd Edition dated March 2002

[12] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (Release 5); reference 3GPP TS 26.234 V5.5.0 (2003-06)

[13] H.264/AVC Standard ISO/IEC 14496-10 (First Edition 2003-12-01): Information technology - Coding of audio-visual objects - Part 10: Advanced video coding with document JVT-K051 "Version 3 of

H.264/AVC" dated 9 June 2004 (errata and Fidelity Range Extensions) and document JVT-L047d8 "Draft Text of H.264/AVC Fidelity Range Extensions Amendment" (AVC Amendment 1 Fidelity Range Extensions, Draft) dated 28 August 2004

[14] MPEG-2 Part 2 (Visual): ISO/IEC 13818-2 Second edition 2000-12-15 (2000 E): Information technology - Generic coding of moving pictures and associated audio information: Video with Amendment 1: Content description data (2001-12-15, corrected version 2002-08-01) and Technical Corrigendum 1 (published 2002-03-01)

[15] MPEG-2 Part 1 (Systems): ISO/IEC 13818-1 Second edition 2000-12-01 (2000 E): Information technology - Generic coding of moving pictures and associated audio information: Systems with Amendment 1: Carriage of metadata over ITU-T Rec. H.222.0 ¦ ISO/IEC 13818-1 streams (2003-08-01, corrected version 2003-10-15) and Technical Corrigendum 1 (published 2002-03-01) and Technical Corrigendum 2 (published 2002-12-01) and Amendment 3 Transport of AVC video data over ITU-T Rec. H222.0/ ISO/IEC 13818-1 streams, dated 2004-11-01

[16] DVD Standard for Video: DVD-Video Book Part 3: Video Specifications v1.1

[17] SMPTE "Proposed SMPTE Standard for Television: VC-1 Compressed Video Bitstream Format and Decoding Process" committee draft 2, revision 1, reference number SMPTE CD xxxM (otherwise referred to as VC-1)

[18] Advanced Systems Format (ASF) Specification: revision 01.20.02; Microsoft Corporation, June 2004

[19] MPEG-2 audio: ISO/IEC 13818-3:1998 Information technology -- Generic coding of moving pictures and associated audio information -- Part 3: Audio

[20] MPEG-2 AAC: ISO/IEC 13818-7:2004 Information technology -- Generic coding of moving pictures and associated audio information -- Part 7: Advanced Audio Coding (AAC)

[21] MPEG-4 AAC (AAC plus): ISO/IEC 14496-3:2001 Information technology -- Coding of audio-visual objects -- Part 3: Audio plus Cor1:2002, Cor1:2004, Cor2:2004, Amd1:2003, Amd2:2004, Amd3

[22] HE AAC codecs: ISO/IEC 14496-3:2001 Information technology -- Coding of audio-visual objects -- Part 3: Audio plus Cor1:2002, Cor1:2004, Cor2:2004, Amd1:2003, Amd2:2004, Amd3

[23] MPEG-1 audio: ISO/IEC 11172-3:1993 Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbit/s – Part3: Audio

[24] MPEG-4 Part 14: ISO/IEC 14496-14:2003 Information technology -- Coding of audio-visual objects -- Part 14: MP4 file format

[25] Digital Audio Compression Standard (AC-3) ATSC A/52, (1995-12)

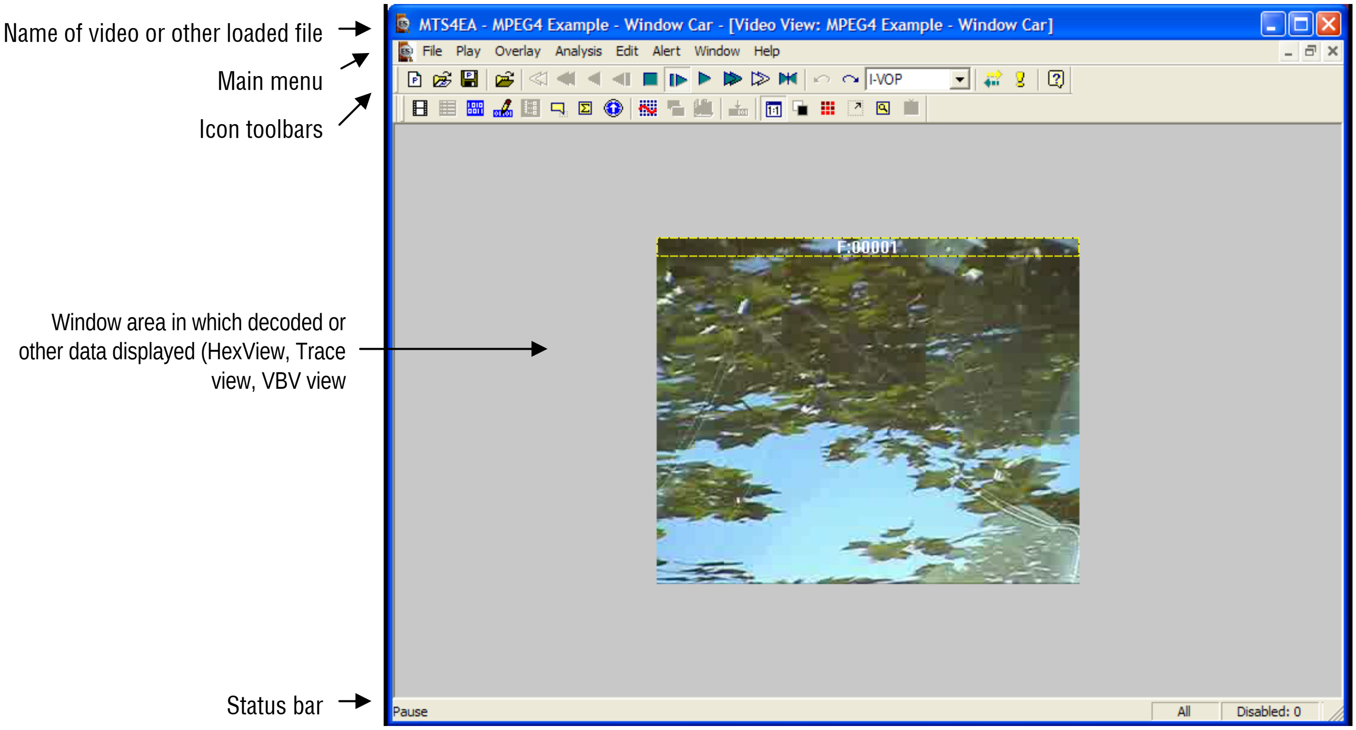
How to Use the MTS4EA

How to Use the MTS4EA

This chapter:

- Describes the elements of MTS4EA display window (see *Window Elements* on page 6–2)
- Tells how to start using MTS4EA display (see *Starting to Use* the on page 6–3)
- Describes each of the menu items in detail and how to use them (see *Main Menu* on page 6–13)
- Explains the icon toolbars, below the main menu (see *Icon Toolbars* on page 6–271)
- Explains the context-sensitive toolbars, which appear depending upon the video standard used and/or the actions being performed (see *Context-sensitive Toolbars/Tooltips* on page 6–275)
- Explains the information provided on the status bar (see *Status Bar* on page 6–278)
- Lists the shortcut keys, such as Ctrl+A to Pause/Frame Advance one frame (see *Ctrl Shortcut Keys* on page 6–280)
- Explains the use of the Alt key to select menu items (see *Alt Menu Keys* on page 6–281)
- Explains the use of the command line/batch mode (see *Command Line/Batch Mode Options* on page 6–286)

Window Elements

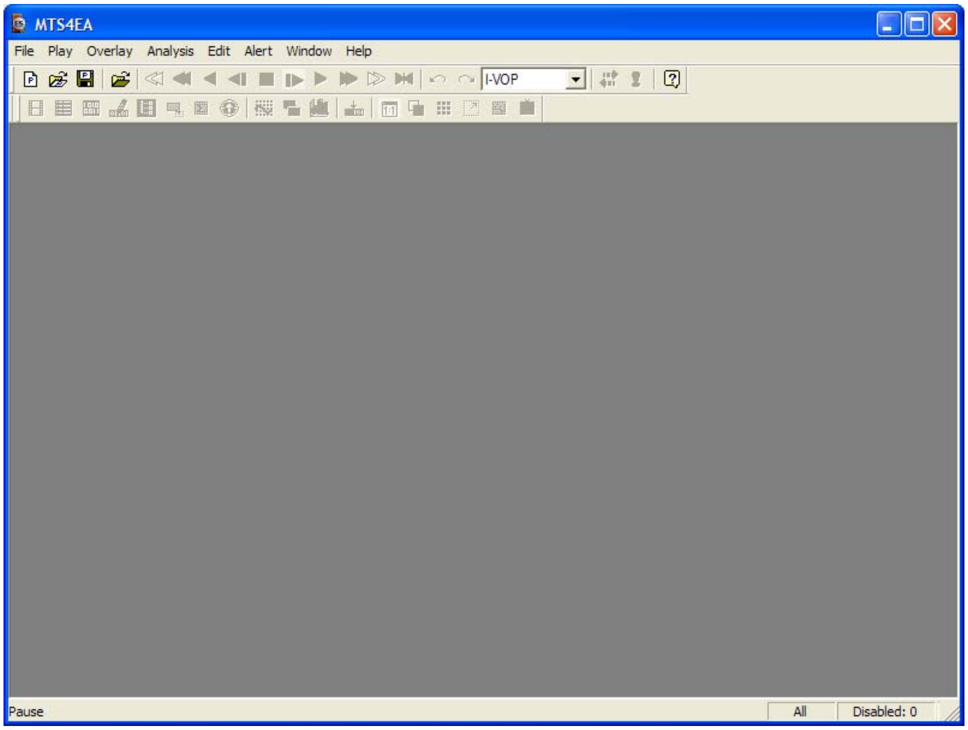

Name of video or other loaded file
Main menu
Icon toolbars
Window area in which decoded or other data displayed (HexView, Trace view, VBV view
Status bar
MTS4EA - MPEG4 Example - Window Car - [Video View: MPEG4 Example - Window Car]
File Play Overlay Analysis Edit Alert Window Help
I-VOP
F:00001
Pause
All
Disabled: 0

NOTE. *The above screenshot was taken with the MTS4EA running under Windows XP. When the MTS4EA is running under Windows 2000, there may be some visual differences in the borders, menus, and the way that popup menus appear from those shown in the screenshots. However, there are no functional differences*
in the waythe MTS4EA operates on these different versions of Windows. (The desktop theme in use will also affect the look.)

In addition to the above toolbars/windows, there are multiple other toolbars that are visible when particular functions are enabled. Also, there are some toolbars, tooltips and menu items that are context-sensitive, depending on the video/audio standard being used and/or the explicit function being performed.

Starting to Use the MTS4EA

When the MTS4EA starts, the display initially looks as below:

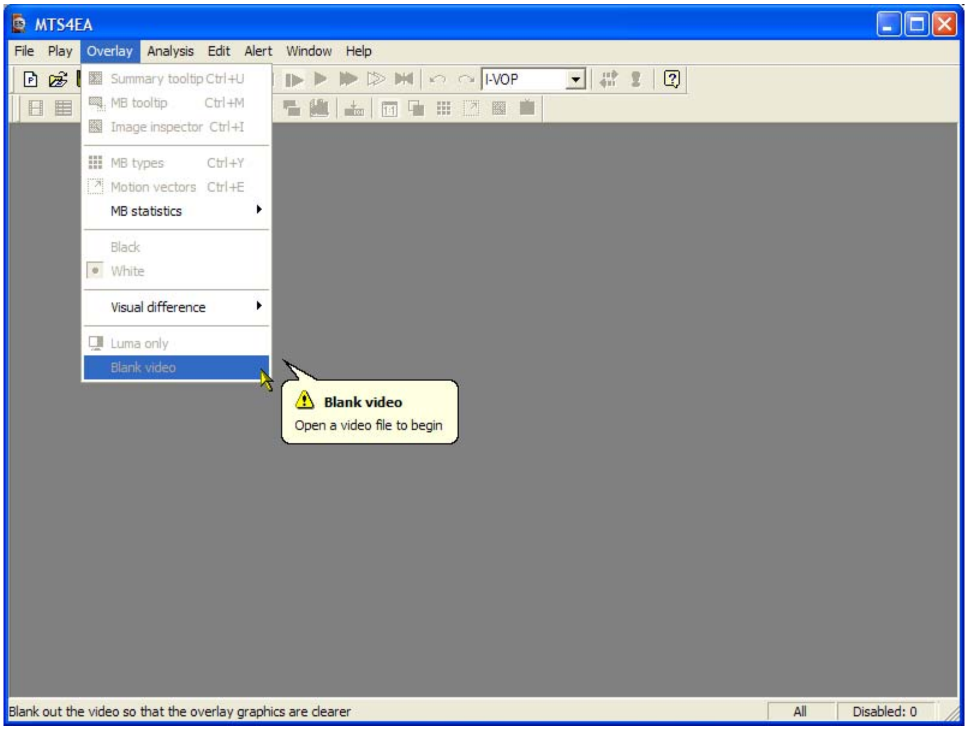


The gray background in the window indicates that no video or other files are loaded. The only icons on the toolbar that are currently enabled are the icons:

- For the Project (New Project, Open Project and Save Project)
- Open stream...
- Help

NOTE. *Some features on some menus may stay permanently grayed out (unavailable). There are two possible reasons for this: the function or feature is never enabled or is not relevant to the standard you are using; or the existing license you have for MTS4EA does not allow all the features to be used* - see MTS4EA License Manager *in chapter 4 for more information.*

When menu items are grayed out, leaving the mouse at the location of the grayed out menu item for a few seconds will display some popup text explaining why the menu item is disabled and/or how it can be enabled. The following example is on the Overlay menu, for the Blank video menu selection:



Blank video
Open a video file to begin

Blank out the video so that the overlay graphics are clearer

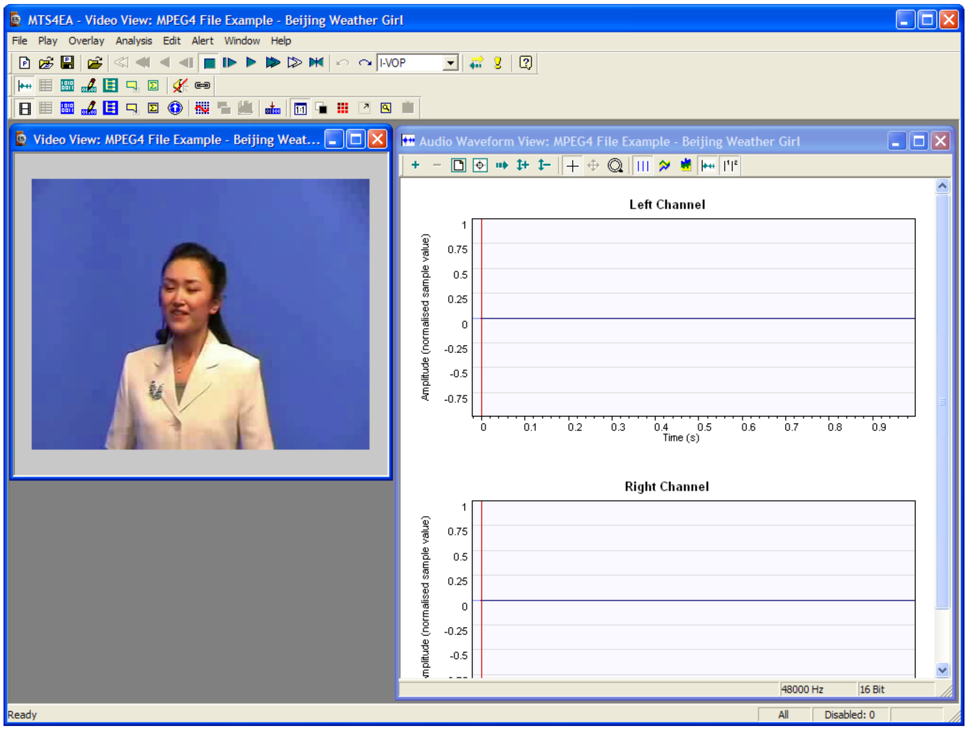
Opening a File

To open a file, do one of the following:

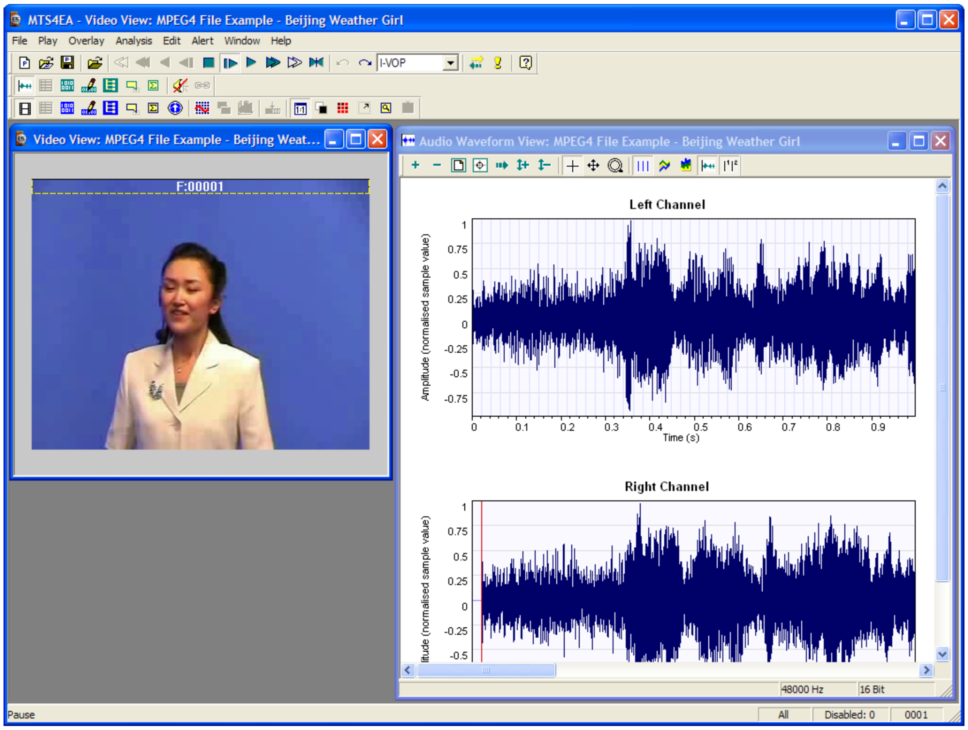
- Click on File > Open stream...
- Click the Open file icon
- Press Ctrl+O

When a stream is loaded, the first frame is decoded and displayed by the MTS4EA. While the first frame of the stream is displayed, the stream itself is not in Play mode, therefore Analysis options can be selected and configured.

In the following screenshot, the example file Beijing Girl is loaded with its first frame displayed, but as indicated by the Stop icon and the empty frame counter in the bottom right corner of the screen, the stream is yet to be played. Where there is audio content, the audio waveform view is also displayed.


MTS4EA - Video View: MPEG4 File Example - Beijing Weather Girl
File Play Overlay Analysis Edit Alert Window Help
I-VOP
Video View: MPEG4 File Example - Beijing Weat...
Audio Waveform View: MPEG4 File Example - Beijing Weather Girl
Left Channel
Amplitude (normalised sample value)
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Time (s)
Right Channel
1
0.75
0.5
0.25
0
-0.25
-0.5
48000 Hz
16 Bit
Ready
All
Disabled: 0

If the loaded stream is initially advanced by one frame, the first frame is again decoded/played before entering the Pause mode. In this mode, Analysis options are not available. When stopped, the first frame reappears on the display until the stream is advanced/played further.


MTS4EA - Video View: MPEG4 File Example - Beijing Weather Girl
File Play Overlay Analysis Edit Alert Window Help
I-VOP
Video View: MPEG4 File Example - Beijing Weat...
F:00001
Audio Waveform View: MPEG4 File Example - Beijing Weather Girl
Left Channel
Amplitude (normalised sample value)
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Time (s)
Right Channel
litude (normalised sample value)
1
0.75
0.5
0.25
0
-0.25
-0.5
48000 Hz
16 Bit
Pause
All
Disabled: 0
0001

As is normal with Windows applications, the window can be dragged around within the frame of the application, resized, maximized, and minimized.

If another file is opened, then this replaces the file last opened; another window is not opened.

NOTE. *However, you can open a number of copies of the MTS4EA to allow direct, side-by-side comparisons of multiple streams.*

Playing Mode: Restrictions

There are two Play modes: these are indicated at the lower left corner of the status bar:

- **Stop**: stream is stopped
- **Play**: stream is paused (after Pause/Step forward) or playing at normal speed, Fast forward, Blind, or Silent fast forward.

Certain actions can be performed only when the open file is in the appropriate Play mode. A few such examples are:

- The Trace enable and Graph enable options on the Analysis menu are only available in Stop mode (see *Video Trace Enable Ctrl+T* on page 6–136 and *Video Graph Enable Ctrl+G* on page 6–153)
- The real-time onscreen image overlays, such as MB types (see *MacroBlock Types* *Ctrl+Y* on page 6–94) and motion vector overlay (see *Motion Vectors* *Ctrl+E* on page 6–98) do not update the video window when in Stop mode

NOTE. *See also* Audio/Video Synchronization *on page 6–51 for further information and details on playing audio and audio-video synchronization.*

MTS4EA Projects

When working with the MTS4EA, the projects feature allows all current setups to be saved at any point and reloaded, for example, to allow work to recommence at exactly the same point. The project file includes information on the name of the file being analyzed, the open views and the location of these, plus many other items. See *New Project...* on page 6–34 for more information.

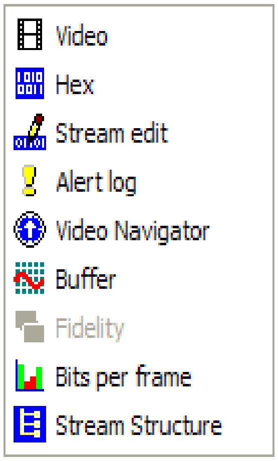
Synchronized Views/Navigating the Views

When a video or audio stream is played, MTS4EA has many views of the video, audio and other data:

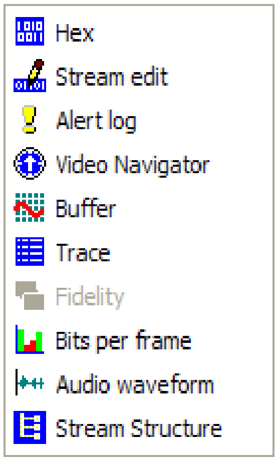
- Alert log
- Video view
- Audio/video hex, stream edit and navigator views
- Buffer analysis view
- Fidelity view (for example PSNR analysis)
- Bits per frame view
- Trace views
- Audio waveform, hex, alert, trace and stream edit views
- File structure view
- Excel graphs

Some of these views are linked. A right-click in one view will open a context menu with a Goto command that will take you to the corresponding location in another view (provided the file being played has both video and audio streams in order to bring up the Audio waveform view).

Example of Goto view menus:



From Trace file



From video view

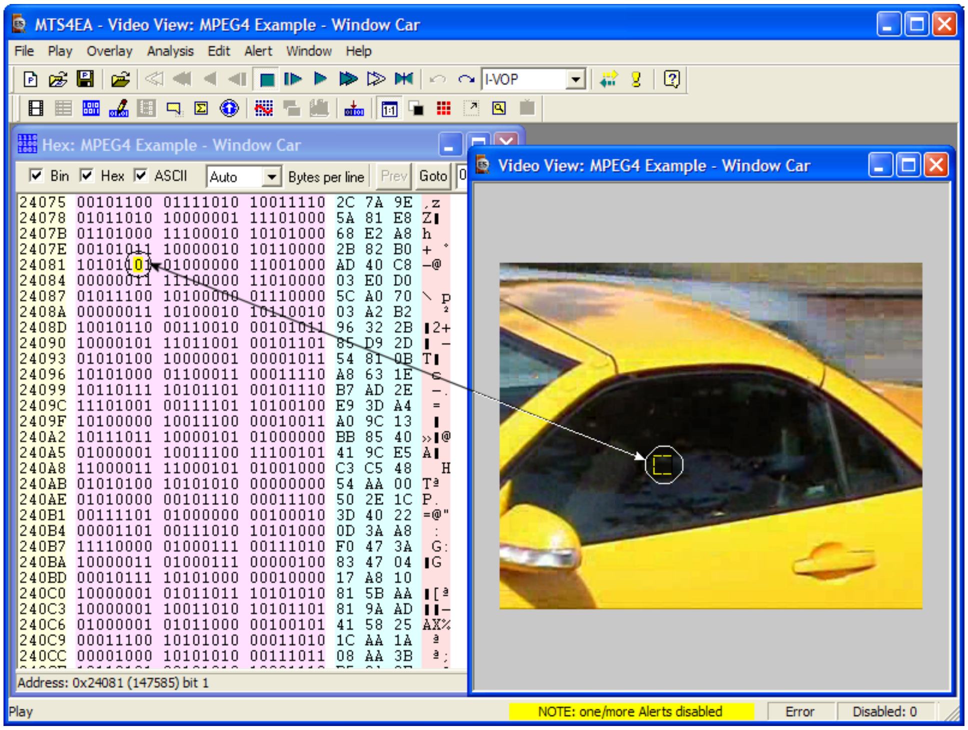
NOTE. *When in the video or audio view, to make it easier to see which area of the window is providing the link to another view, there is always a small square or rectangle in the view that highlights the MacroBlock or frame for cross-referencing in the other view.*

Example of moving between views. For example, when in the video view and using the MacroBlock tooltip, to see the hex/binary data for a specific MacroBlock, right-click at the MacroBlock location of interest and select Goto view from the context menu, and then select the HexView. The HexView window opens (if it is not already opened for this stream) and the first byte of the selected MacroBlock is highlighted.

Selecting a specific byte in the HexView, and then using the right-click context menu to select the Video view will take the video window to the MacroBlock that contains the selected byte, with the MacroBlock highlighted (see *Highlighting of selected areas in the video view* on page 6–9).

Highlighting of selected areas in the video view. If a data area is selected in a view other than the video view, and the context menu is used to go to the video view, the corresponding area of the video view is highlighted, as follows:

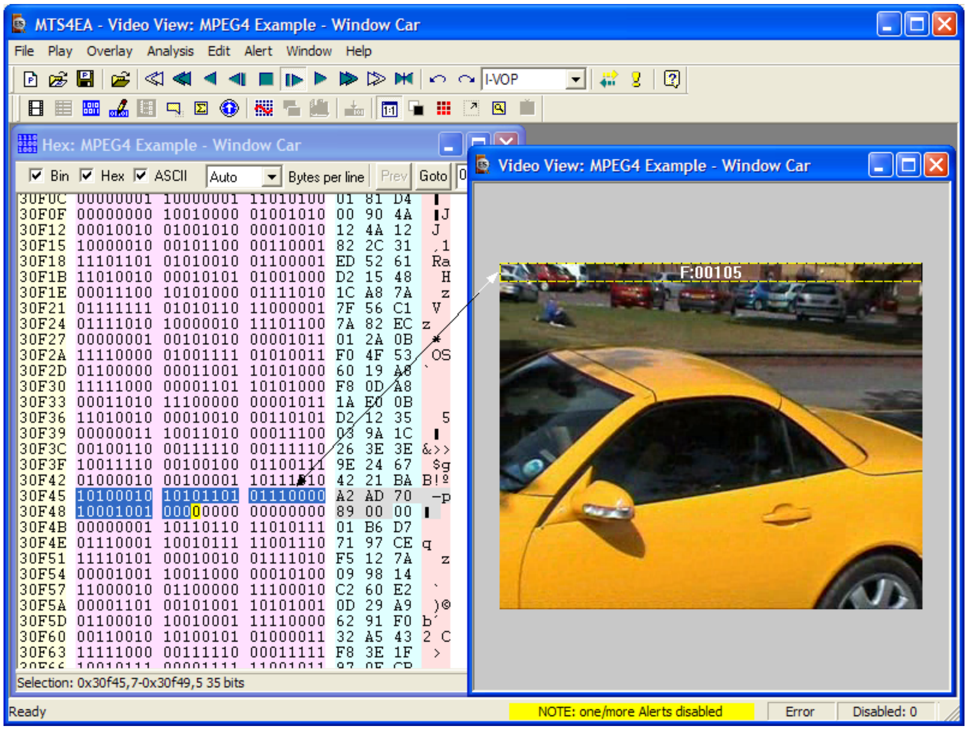
- If the area can be resolved to a single MacroBlock, then a yellow and black dotted box is displayed around the MacroBlock concerned:


MTS4EA - Video View: MPEG4 Example - Window Car
File Play Overlay Analysis Edit Alert Window Help
I-VOP
Hex: MPEG4 Example - Window Car
☑ Bin ☑ Hex ☑ ASCII
Auto
Bytes per line
Prev
Goto
Video View: MPEG4 Example - Window Car
Play
NOTE: one/more Alerts disabled
Error
Disabled: 0

```
24075 00101100 01111010 10011110 2C 7A 9E ,z
24078 01011010 10000001 11101000 5A 81 E8 Z▌
2407B 01101000 11100010 10101000 68 E2 A8 h
2407E 00101011 10000010 10110000 2B 82 B0 + °
24081 10101101 01000000 11001000 AD 40 C8 –@
24084 00000011 11100000 11010000 03 E0 D0
24087 01011100 10100000 01110000 5C A0 70 \ p
2408A 00000011 10100010 10110010 03 A2 B2   ²
2408D 10010110 00110010 00101011 96 32 2B ▌2+
24090 10000101 11011001 00101101 85 D9 2D ▌ –
24093 01010100 10000001 00001011 54 81 0B T▌
24096 10101000 01100011 00011110 A8 63 1E  c
24099 10110111 10101101 00101110 B7 AD 2E  –.
2409C 11101001 00111101 10100100 E9 3D A4  =
2409F 10100000 10011100 00010011 A0 9C 13  ▌
240A2 10111011 10000101 01000000 BB 85 40 »▌@
240A5 01000001 10011100 11100101 41 9C E5 A▌
240A8 11000011 11000101 01001000 C3 C5 48   H
240AB 01010100 10101010 00000000 54 AA 00 Tª
240AE 01010000 00101110 00011100 50 2E 1C P.
240B1 00111101 01000000 00100010 3D 40 22 =@"
240B4 00001101 00111010 10101000 0D 3A A8  :
240B7 11110000 01000111 00111010 F0 47 3A  G:
240BA 10000011 01000111 00000100 83 47 04 ▌G
240BD 00010111 10101000 00010000 17 A8 10
240C0 10000001 01011011 10101010 81 5B AA ▌[ª
240C3 10000001 10011010 10101101 81 9A AD ▌▌–
240C6 01000001 01011000 00100101 41 58 25 AX%
240C9 00011100 10101010 00011010 1C AA 1A  ª
240CC 00001000 10101010 00111011 08 AA 3B  ª;
```

Address: 0x24081 (147585) bit 1

- If the area cannot be resolved to a single MacroBlock, then a yellow and black dotted rectangle is displayed surrounding the top row of MacroBlocks in the video frame that contains the first selected area:


MTS4EA - Video View: MPEG4 Example - Window Car
File
Play
Overlay
Analysis
Edit
Alert
Window
Help
I-VOP
Hex: MPEG4 Example - Window Car
☑ Bin
☑ Hex
☑ ASCII
Auto
Bytes per line
Prev
Goto
Video View: MPEG4 Example - Window Car
F:00105

```
30F0C 00000001 10000001 11010100 01 81 D4
30F0F 00000000 10010000 01001010 00 90 4A  J
30F12 00010010 01001010 00010010 12 4A 12  J
30F15 10000010 00101100 00110001 82 2C 31  ,1
30F18 11101101 01010010 01100001 ED 52 61  Ra
30F1B 11010010 00010101 01001000 D2 15 48   H
30F1E 00011100 10101000 01111010 1C A8 7A   z
30F21 01111111 01010110 11000001 7F 56 C1  V
30F24 01111010 10000010 11101100 7A 82 EC z
30F27 00000001 00101010 00001011 01 2A 0B  *
30F2A 11110000 01001111 01010011 F0 4F 53  OS
30F2D 01100000 00011001 10101000 60 19 A8 `
30F30 11111000 00001101 10101000 F8 0D A8
30F33 00011010 11100000 00001011 1A E0 0B
30F36 11010010 00010010 00110101 D2 12 35   5
30F39 00000011 10011010 00011100 03 9A 1C
30F3C 00100110 00111110 00111110 26 3E 3E &>>
30F3F 10011110 00100100 01100111 9E 24 67  $g
30F42 01000010 00100001 10111010 42 21 BA B!º
30F45 10100010 10101101 01110000 A2 AD 70  -p
30F48 10001001 00000000 00000000 89 00 00
30F4B 00000001 10110110 11010111 01 B6 D7
30F4E 01110001 10010111 11001110 71 97 CE q
30F51 11110101 00010010 01111010 F5 12 7A   z
30F54 00001001 10011000 00010100 09 98 14
30F57 11000010 01100000 11100010 C2 60 E2  `
30F5A 00001101 00101001 10101001 0D 29 A9  )©
30F5D 01100010 10010001 11110000 62 91 F0 b´
30F60 00110010 10100101 01000011 32 A5 43 2 C
30F63 11111000 00111110 00011111 F8 3E 1F  >
```

Selection: 0x30f45,7-0x30f49,5 35 bits

Ready   NOTE: one/more Alerts disabled   Error   Disabled: 0

- If the selected area is in a frame or file header, not within a specific MacroBlock, then a yellow and black dotted rectangle is displayed surrounding the top row of MacroBlocks in the video frame that immediately follows the frame or file header (see the preceding illustration: the data selected in the HexView window is in the VOP header, so the highlight rectangle is displayed across the top row of MacroBlocks of the corresponding VOP).

When the Black/White digits icon is pushed in, the video plays in black and white.

Highlighting of selected areas in the audio view. Highlighting selected areas in the audio waveform view works in a similar manner to highlighting areas in the video view (see page 6–9).

NOTE. *An audio frame consists of a fixed number (per channel) of audio samples representing a fixed time length of decoded audio. Typically each iteration through the decode loop will output one audio frame of decoded audio.*

Synchronization of views/Synchronize views icon . When the Synchronize views icon is pushed in, then all open windows automatically follow the selection that you made.

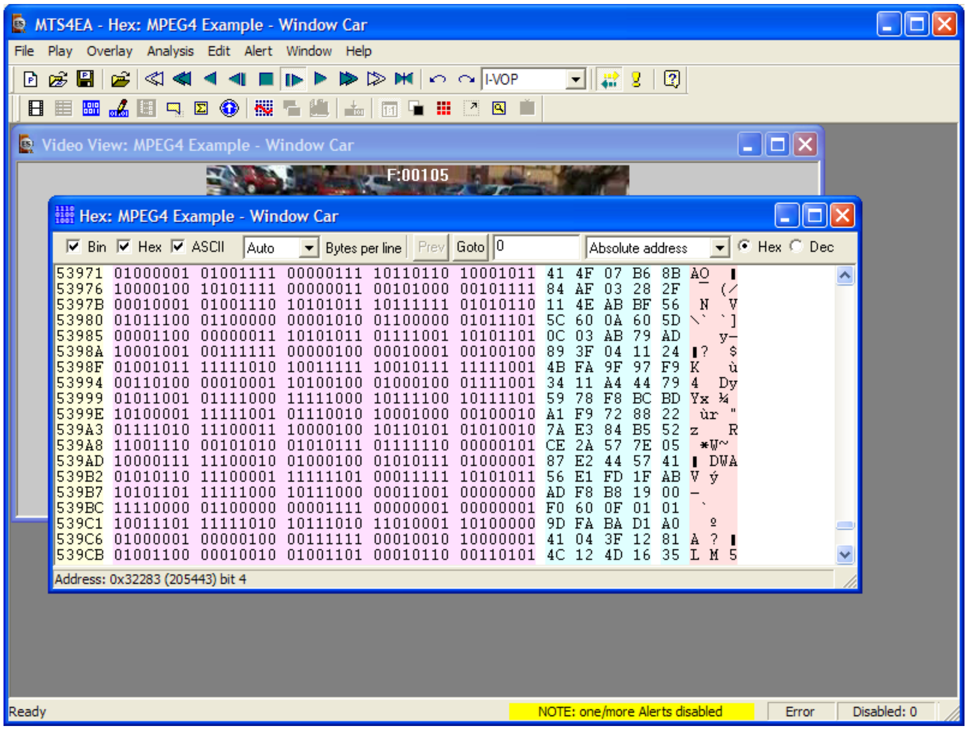
For example, if the Video Navigator and Hex views are both open at the same time as the Audio waveform view, and the Synchronize views icon is pushed in, and video is then played, all views display the same relative points in the stream at the same time.

NOTE. *This can cause a delay when playing video for some views (such as the Trace views) or when the memory buffers used by the MTS4EA do not contain all the required data and therefore time is taken to decode and interpret the video and refill the buffers.*

Opening Multiple Windows

Within the MTS4EA it is possible to open many windows at once. These can be:

- A video/audio window with another window, such as a HexView window with a video window (as shown in the following figure):


MTS4EA - Hex: MPEG4 Example - Window Car
File
Play
Overlay
Analysis
Edit
Alert
Window
Help
I-VOP
Video View: MPEG4 Example - Window Car
F:00105
Hex: MPEG4 Example - Window Car
Prev
Goto

☑ Bin ☑ Hex ☑ ASCII Auto Bytes per line Goto 0 Absolute address ● Hex ○ Dec

```
53971 01000001 01001111 00000111 10110110 10001011 41 4F 07 B6 8B AO
53976 10000100 10101111 00000011 00101000 00101111 84 AF 03 28 2F   (/
5397B 00010001 01001110 10101011 10111111 01010110 11 4E AB BF 56  N  V
53980 01011100 01100000 00001010 01100000 01011101 5C 60 0A 60 5D \` `]
53985 00001100 00000011 10101011 01111001 10101101 0C 03 AB 79 AD    y-
5398A 10001001 00111111 00000100 00010001 00100100 89 3F 04 11 24 ?  $
5398F 01001011 11111010 10011111 10010111 11111001 4B FA 9F 97 F9 K   ù
53994 00110100 00010001 10100100 01000100 01111001 34 11 A4 44 79 4  Dy
53999 01011001 01111000 11111000 10111100 10111101 59 78 F8 BC BD Yx ¼
5399E 10100001 11111001 01110010 10001000 00100010 A1 F9 72 88 22  ùr "
539A3 01111010 11100011 10000100 10110101 01010010 7A E3 84 B5 52 z   R
539A8 11001110 00101010 01010111 01111110 00000101 CE 2A 57 7E 05  *W~
539AD 10000111 11100010 01000100 01010111 01000001 87 E2 44 57 41   DWA
539B2 01010110 11100001 11111101 00011111 10101011 56 E1 FD 1F AB V ý
539B7 10101101 11111000 10111000 00011001 00000000 AD F8 B8 19 00 -
539BC 11110000 01100000 00001111 00000001 00000001 F0 60 0F 01 01  `
539C1 10011101 11111010 10111010 11010001 10100000 9D FA BA D1 A0    º
539C6 01000001 00000100 00111111 00010010 10000001 41 04 3F 12 81 A ?
539CB 01001100 00010010 01001101 00010110 00110101 4C 12 4D 16 35 L M 5
```

Address: 0x32283 (205443) bit 4

Ready   NOTE: one/more Alerts disabled   Error   Disabled: 0

- Other combinations, such as the video/audio and HexView with the Trace file view and file structure view.

To open windows other than for video files (or files that contain video, such as MP4 or 3GPP files, or MPEG-2 Program/Transport Streams):

- For Trace views, do one of the following:
    - Select File > Open other... (or click the icon) to open a Trace file previously stored or associated with another video stream
    - Select Analysis > View trace... (or Ctrl+V) to open the current trace file, as named in the Analysis menu - Trace tab
- To view the statistical graphs (in Excel), click on Analysis > View graphs...
- For HexView, click on Analysis > View stream hex... (or click the icon or press Ctrl+H)
- For the stream structure view, click on Analysis > View file structure... (or click the icon or press Ctrl+R)
- For the buffer analysis view, click on Analysis > View buffer analysis... (or click the icon)
- For the fidelity analysis, click on Analysis > View fidelity analysis... (or click the icon)
- For the Alert Log, click on Alert > View alert log... (or click the icon)
- For the Audio waveform view, click on the icon . (If there is audio content, the toolbar and the view will be displayed automatically when the video view is opened.)

NOTE. *The Audio waveform view is available only when an audio stream or file/project containing an audio stream or streams is open.*

Main Menu

The submenu options available under each main menu selection (for example, File, Play, Overlay) are described in detail in their respective sections. A summary of each main menu item is given here.

Several of the common functions can be accessed through:

- The icon toolbars (see *Icon Toolbars* on page 6–271)
- Ctrl keys (see *Ctrl Shortcut Keys* on page 6–280)

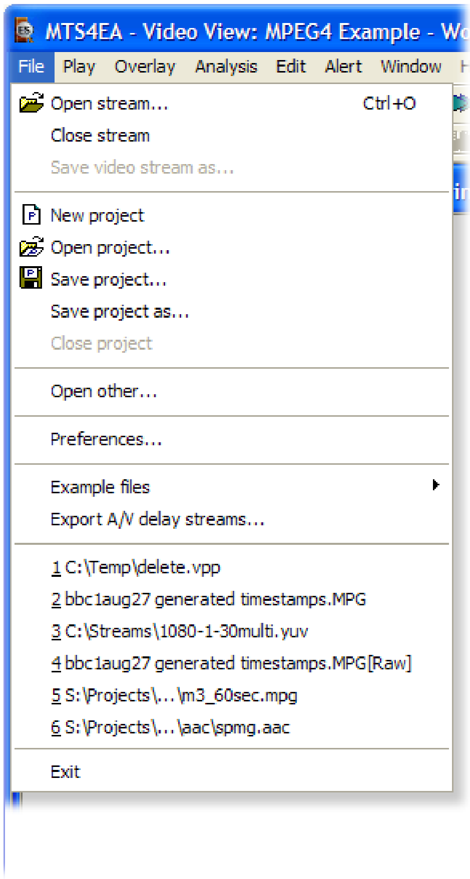
NOTE. *You can open a number of copies of MTS4EA to allow direct, side-by-side comparisons of multiple video streams.*

Summary of Function of each Main Menu Item

Main menu item	Principal functions/sub-options
File	Opening and saving files and projects, recent file list, example files, exit
Play	Play video and audio, stop, pause, fast forward, set decoder options
Overlay	Hide/show video/audio data and tooltips
Analysis	Set Trace and Graph options; view Trace, Graph files, Hex view, file structures, buffer analysis, fidelity analysis
Edit	Edit elementary video/audio stream, edit bitstream selection, save edited stream
Alert	Enable/disable/configure overall alert levels and individual alerts
Window	Arrange windows and toolbars, view settings, synchronize views, window list
Help	Help topics, PDF help and tutorials, license manager, version info

File Menu

The following paragraphs describe the options available in the File menu.


MTS4EA - Video View: MPEG4 Example - Wo
File
Play
Overlay
Analysis
Edit
Alert
Window
Open stream...
Ctrl+O
Close stream
Save video stream as...
New project
Open project...
Save project...
Save project as...
Close project
Open other...
Preferences...
Example files
Export A/V delay streams...
1 C:\Temp\delete.vpp
2 bbc1aug27 generated timestamps.MPG
3 C:\Streams\1080-1-30multi.yuv
4 bbc1aug27 generated timestamps.MPG[Raw]
5 S:\Projects\...\m3_60sec.mpg
6 S:\Projects\...\aac\spmg.aac
Exit

Open Stream... Ctrl+O

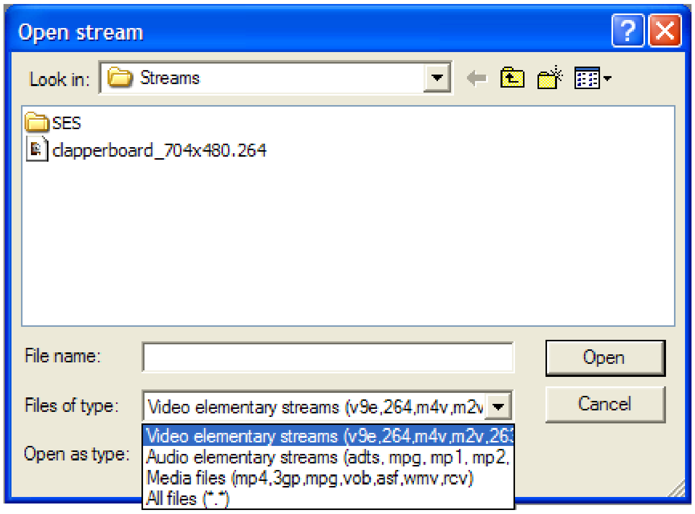
With this option, you can browse the file system to find a file to analyze. This file can be:

- A file containing only audio, for example, an MPG file
- A compressed video file that contains video data only, for example, an MPEG-4 video Elementary Stream (see *Compression Standards and File Types*).
- A container file that includes audio and transport information as well as video, for example MP4, 3GPP and MPEG-2 (see *Compression Standards and File Types*).
- An uncompressed video file, see *Opening an Uncompressed Video File (any file extension)* on page 6–27.

Autodetect File Type

When Autodetect is selected in the Open as type drop-down box, the file contents are automatically selected by the MTS4EA.

In this case, the Files of type drop-down is simply used to narrow the list of file extensions searched for:



Open stream

Look in: Streams

SES
clapperboard_704x480.264

File name:

Files of type: Video elementary streams (v9e,264,m4v,m2v

Video elementary streams (v9e,264,m4v,m2v,263
Audio elementary streams (adts, mpg, mp1, mp2,
Media files (mp4,3gp,mpg,vob,asf,wmv,rcv)
All files (*.*)

Open as type:

The default file extensions that the MTS4EA looks for are:

- For video-only files: .v9e, .264, .m4v, .m2v, .263, .261, .cmp, .bits

- For audio-only files: .adts, .mpg, .mp1, mp2, aac, ac3
- For container or media files including video: .mp4, .3gp, .mpg, .vob, .asf, .wmv, .rcv

With Autodetect selected, whichever of the file types/extensions is selected, the MTS4EA will automatically override the selection and determine the standard that the file is compressed to.

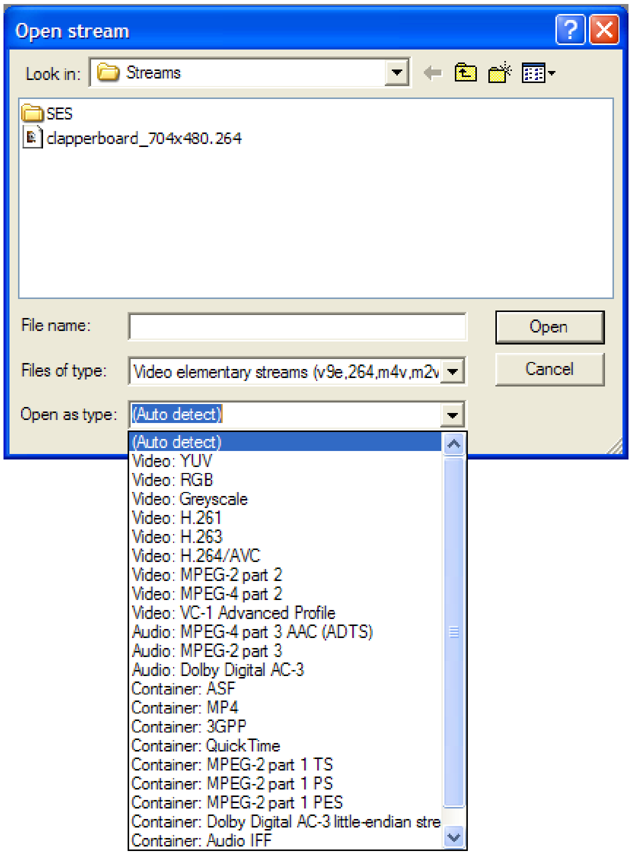
NOTE. *The exceptions to this autodetection are VOB files, where MTS4EA will take different actions due to the VOB extension (see* Open Stream... 📂 `Ctrl+O` *on page 6–16).*

The selected file extension is remembered by the MTS4EA for the next time a file is opened.

Force to Open as a Specific File Type

Sometimes, particularly if there are errors in a file, the autodetect function of the MTS4EA does not correctly identify the type of data contained in the file.

In this case, the file type can be forced by selecting a file type from the drop-down Open as type list. In this case, the file extension is ignored.



Open stream

Look in: Streams

SES
clapperboard_704x480.264

File name:

Files of type: Video elementary streams (v9e,264,m4v,m2v

Open as type: (Auto detect)

NOTE. *To open an uncompressed video file, select the appropriate item from the list and the MTS4EA will open the file in the appropriate format, regardless of the file extension or the data found in the file.*

Opening a Video/Audio Elementary Stream

When a raw video/audio elementary stream file is opened, initial checks are done on the file for validity and to determine the compression standard.

When the file has been read in correctly, the appearance of the Play icon on the toolbar changes from grayed out ▷ to blue ▶, to indicate that the file is ready to play.

If MTS4EA recognizes that the open file includes an audio stream, the toolbar is displayed.

- Opens the Audio waveform view (see Show Audio waveform ( ), page 6–124).
- 🔇 Turns the sound off/on: the stream continues to play and decode even when sound is disabled (see Mute audio 🔇 on page 6–69).

If the file contains only audio data, the video toolbar will not be displayed. The Synchronize audio icon is also unavailable, as there is no video with which to synchronize (see *Audio/Video Synchronization* on page 6–51).

The title of the elementary stream is displayed in the title bar of the video and audio window.

Opening a Container File

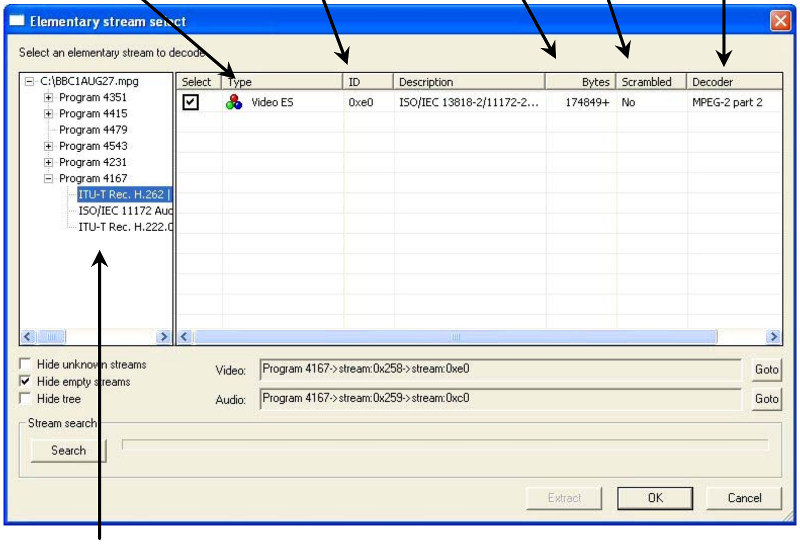
Track type (video, audio, other – note the icon)

Track/stream ID number

Size of track/stream

Scrambled (encrypted)

Decoder type (see below)



Elementary stream select

Select an elementary stream to decode

- C:\BBC1AUG27.mpg
  - Program 4351
  - Program 4415
  - Program 4479
  - Program 4543
  - Program 4231
  - Program 4167
    - ITU-T Rec. H.262 |
    - ISO/IEC 11172 Auc
    - ITU-T Rec. H.222.0

Select	Type	ID	Description	Bytes	Scrambled	Decoder
☑	Video ES	0xe0	ISO/IEC 13818-2/11172-2...	174849+	No	MPEG-2 part 2

☐ Hide unknown streams
☑ Hide empty streams
☐ Hide tree

Video: Program 4167->stream:0x258->stream:0xe0 Goto

Audio: Program 4167->stream:0x259->stream:0xc0 Goto

Stream search

Search

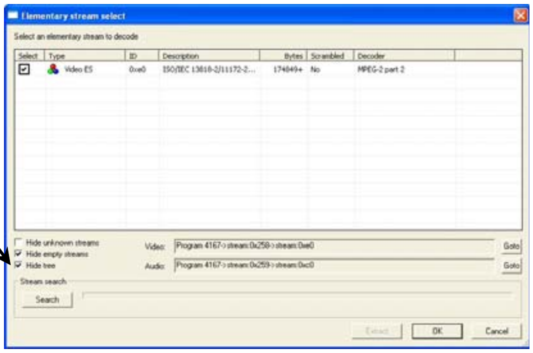
Extract OK Cancel

Tree structure showing the various programs in the file (click the + and – to open and close the items)

Streams selected for analysis – enter check mark in Select column

Streams selected for analysis – enter check mark in Select column

Check box below to hide tree structure



Elementary stream select

Select an elementary stream to decode

Select	Type	ID	Description	Bytes	Scrambled	Decoder
☑	Video ES	0xe0	ISO/IEC 13818-2/11172-2...	174849+	No	MPEG-2 part 2

☐ Hide unknown streams
☑ Hide empty streams
☑ Hide tree

Video: Program 4167->stream:0x258->stream:0xe0 Goto

Audio: Program 4167->stream:0x259->stream:0xc0 Goto

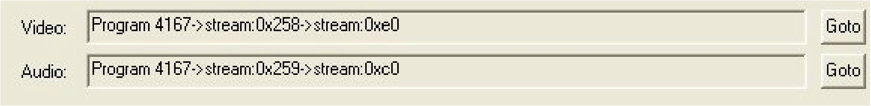
Stream search

Search

Extract OK Cancel

When a container file is loaded, the MTS4EA automatically opens the Elementary stream select window (as shown above), in which are listed all streams, audio and video. The streams contained can be browsed, individually selected for analysis, with your choice of decoder, and either immediately played and analyzed within the MTS4EA or extracted and saved to a file before going on to play/analyze them.

Select. To select the stream, audio or video, for analysis, enter a check mark in this column beside the desired stream. Its description appears in the Video or Audio fields under the browsing area (as shown in the following screenshot). Only one video and one audio stream can be selected; if a stream is selected and a check mark is subsequently entered beside another stream of the same type, the description will change to show the updated selection.



Video: Program 4167->stream:0x258->stream:0xe0

Audio: Program 4167->stream:0x259->stream:0xc0

Clicking on Goto will move the view in the browser window to the location in the tree structure containing that stream, where it can be deselected or the Decoder can be chosen from the drop-down menu (see below).

Type. This is the media stream type as identified by the container signaling information. If insufficient signaling information is present, this may show a general type such as Video ES (Elementary Stream), Audio ES or, where the type is unknown, no text at all.

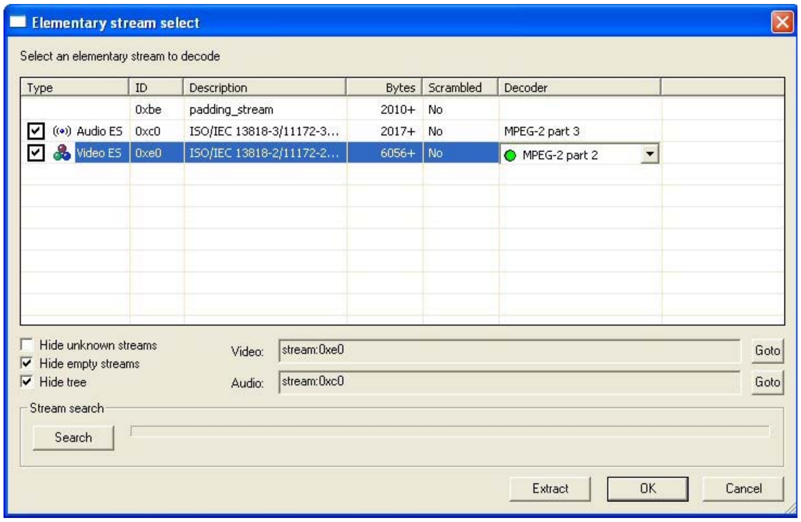
ID. This is the media-stream identifier as dictated by the container signaling information, using appropriate notation for the container standard.

Description. This is a description of the media stream as dictated by the container signaling information, incorporating text from the relevant standard.

Bytes. This is the number of bytes that will be extracted for the stream. If the value is suffixed by a + sign, the total stream size is not known; without the + sign, this value is the final total.

Scrambled. This column shows Yes if the stream is scrambled or encrypted in some way. Scrambled streams cannot be decoded.

Decoder. Once a stream is selected, the decoder must be chosen. The Decoder column shows the decoder(s) found for the selected stream. There can be multiple streams containing video or audio, each with different decoders. If the Decoder column is empty, then the MTS4EA has been unable to match one of its decoders with the data it found.



Elementary stream select

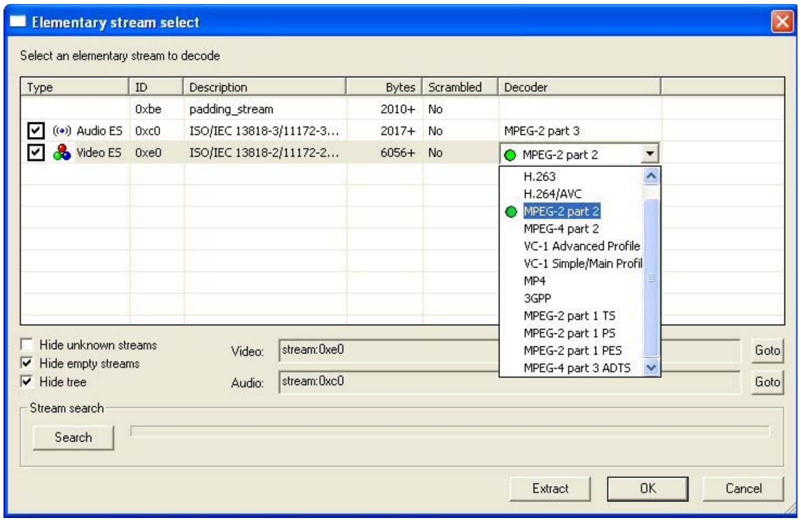
Select an elementary stream to decode

Type	ID	Description	Bytes	Scrambled	Decoder
	0xbe	padding_stream	2010+	No	
☑ Audio ES	0xc0	ISO/IEC 13818-3/11172-3...	2017+	No	MPEG-2 part 3
☑ Video ES	0xe0	ISO/IEC 13818-2/11172-2...	6056+	No	● MPEG-2 part 2

□ Hide unknown streams
☑ Hide empty streams
☑ Hide tree

Video: stream:0xe0
Audio: stream:0xc0

Clicking on the drop-down arrow shows a list of all the available decoders, with the decoder that has been autodetected shown by a green circle:



Elementary stream select

Select an elementary stream to decode

Type	ID	Description	Bytes	Scrambled	Decoder
	0xbe	padding_stream	2010+	No	
☑ Audio ES	0xc0	ISO/IEC 13818-3/11172-3...	2017+	No	MPEG-2 part 3
☑ Video ES	0xe0	ISO/IEC 13818-2/11172-2...	6056+	No	● MPEG-2 part 2

□ Hide unknown streams
☑ Hide empty streams
☑ Hide tree

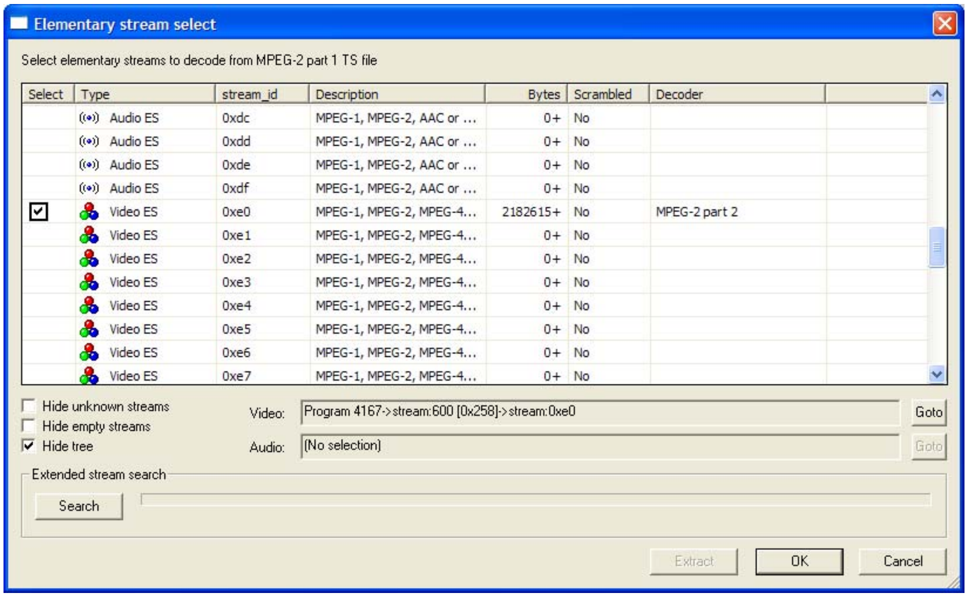
Video: stream:0xe0
Audio: stream:0xc0

Selecting a different decoder forces the MTS4EA to use that decoder for this track/Elementary Stream.

Hide unknown streams. When selected, Hide unknown streams means that the streams are not shown for which the MTS4EA has been unable to match one of its decoders.

When Hide unknown streams is enabled, the empty streams are also hidden, because they will not contain valid data for which the MTS4EA can select a decoder.

Hide empty streams. By default, this is enabled. When enabled, this hides streams that appear to be 0 bytes in size (note that these can still be searched to see if they are in fact larger - see below, under Search).



Select elementary streams to decode from MPEG-2 part 1 TS file

Select	Type	stream_id	Description	Bytes	Scrambled	Decoder
	Audio ES	0xdc	MPEG-1, MPEG-2, AAC or ...	0+	No	
	Audio ES	0xdd	MPEG-1, MPEG-2, AAC or ...	0+	No	
	Audio ES	0xde	MPEG-1, MPEG-2, AAC or ...	0+	No	
	Audio ES	0xdf	MPEG-1, MPEG-2, AAC or ...	0+	No	
☑	Video ES	0xe0	MPEG-1, MPEG-2, MPEG-4...	2182615+	No	MPEG-2 part 2
	Video ES	0xe1	MPEG-1, MPEG-2, MPEG-4...	0+	No	
	Video ES	0xe2	MPEG-1, MPEG-2, MPEG-4...	0+	No	
	Video ES	0xe3	MPEG-1, MPEG-2, MPEG-4...	0+	No	
	Video ES	0xe4	MPEG-1, MPEG-2, MPEG-4...	0+	No	
	Video ES	0xe5	MPEG-1, MPEG-2, MPEG-4...	0+	No	
	Video ES	0xe6	MPEG-1, MPEG-2, MPEG-4...	0+	No	
	Video ES	0xe7	MPEG-1, MPEG-2, MPEG-4...	0+	No	

☐ Hide unknown streams
☐ Hide empty streams
☑ Hide tree

Video: Program 4167->stream:600 [0x258]->stream:0xe0

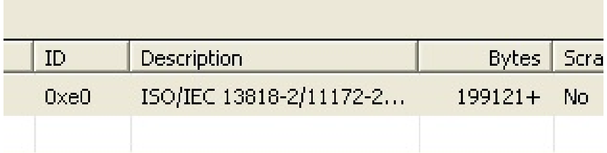
Audio: [No selection]

Search (Extended stream search). For some standards, the total size of the stream is indicated in the container file - for example, MPEG-4. In this case, the size of the elementary stream is shown.

However, for some container formats, the size of the elementary streams is not indicated in the container file, and the only way to know the total size is by searching the entire file. (An example of this is MPEG-2 VOBs.)

For these types of files, the MTS4EA reads the first part of the file only, until an MTS4EA decoder can be matched with the data found.

In this case, the MTS4EA will display the size in the Bytes column with a + sign to the right of the size, as shown in the following figure:



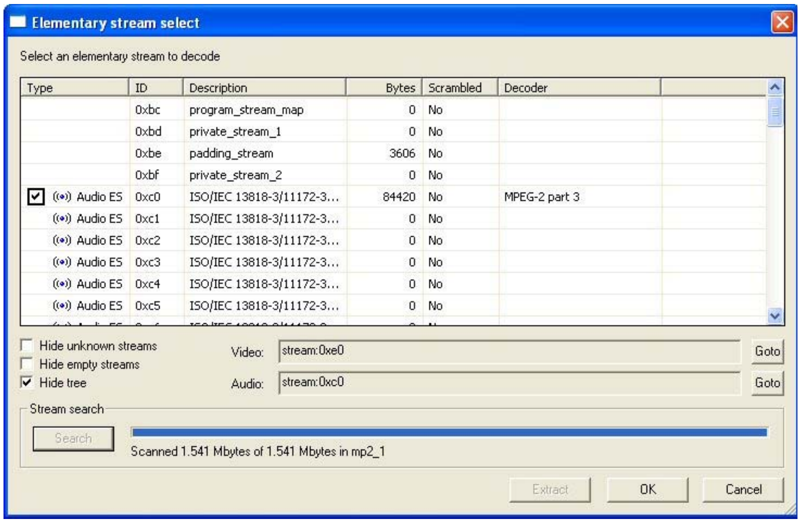
	ID	Description	Bytes	Scra
	0xe0	ISO/IEC 13818-2/11172-2...	199121+	No

The + sign indicates that the whole file has not been searched and that there may be additional data.

NOTE. *This stream selection also means that other views, such as the HexView, may not show the data from the whole file (in which case, a warning is shown). See* View Video Stream Hex... `Ctrl+H` *on page 6–176 for more information.*

In order to find the total size of the whole file, and to search other streams, use the Search button. All streams, including the highlighted, or selected, stream, are then searched.

When the search is complete, the display will be similar to the following figure:



Select an elementary stream to decode

Type	ID	Description	Bytes	Scrambled	Decoder
	0xbc	program_stream_map	0	No	
	0xbd	private_stream_1	0	No	
	0xbe	padding_stream	3606	No	
	0xbf	private_stream_2	0	No	
☑ Audio ES	0xc0	ISO/IEC 13818-3/11172-3...	84420	No	MPEG-2 part 3
Audio ES	0xc1	ISO/IEC 13818-3/11172-3...	0	No	
Audio ES	0xc2	ISO/IEC 13818-3/11172-3...	0	No	
Audio ES	0xc3	ISO/IEC 13818-3/11172-3...	0	No	
Audio ES	0xc4	ISO/IEC 13818-3/11172-3...	0	No	
Audio ES	0xc5	ISO/IEC 13818-3/11172-3...	0	No	

□ Hide unknown streams
□ Hide empty streams
☑ Hide tree

Video: stream:0xe0

Audio: stream:0xc0

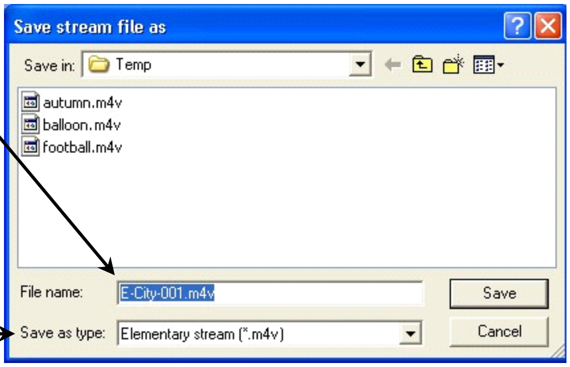
Stream search

Scanned 1.541 Mbytes of 1.541 Mbytes in mp2_1

During the search, the blue progress bar fills, until the search is complete.

During the search process, the MTS4EA will search through all the linked files, and therefore may find other streams. It will be able to determine the total size of the streams found. The result is that the additional streams (and their sizes) will be shown in the stream window.

Extract. This allows the highlighted track, audio or video, to be saved as a separate elementary stream (in the case below, .m4v is offered as the file type, as the container file was MP4):



File name: E-City-001.m4v

Save as type: Elementary stream (*.m4v)

The video track ID from the MP4/3GPP file is appended to the name of the MP4/3CPP file name

A file extension is added automatically, depending on the video type (see below)

NOTE. *If you decide to play and analyze the selected stream immediately, and then decide to save it later, this can be done by clicking the* Save as... *option on the* File *menu (see* Save Video Stream *As… on page 6–33).*

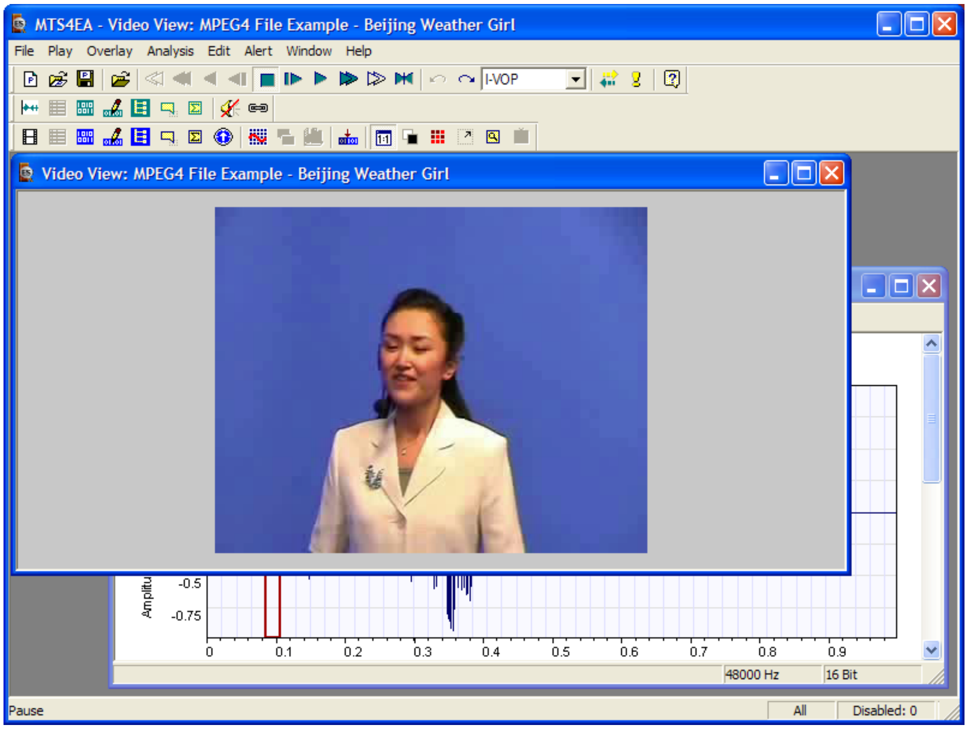
The correct type is automatically added as a file extension, depending upon the data in the file:

- H.264/AVC video format files are given the extension: `.264`
- VC-1 Advanced Profile video files are given the extension: `.v9e`
- MPEG-4 video format files are given the extension: `.m4v`
  (except for MPEG-4 Short Header, with extension `.263`)
- MPEG-2 video format files are given the extension: `.m2v`
- H.263, H.263+ and MPEG-4 Short Header video format files are given the extension: `.263`

NOTE. *The suggested file name and extension do not have to be accepted - any file name and extension can be entered.*

OK. OK opens the selected stream using the decoder shown; if the OK button is grayed out, then the selected stream cannot be opened.

A window is opened with the title of the container file with the track/stream number.



Because the MTS4EA has recognized that the open file contains both audio and video streams, the video and the audio views are opened. The audio toolbar is also displayed and the synchronize is enabled (see Synchronize audio , page 6–69).

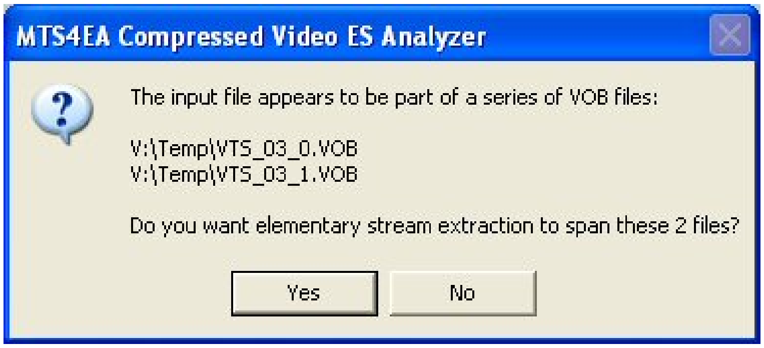
NOTE. *If you decide to play and analyze the selected stream immediately, and then decide to save it later, this can be done by clicking the Save as... option on the File menu (see* Save Video Stream *As… on page 6–33).*

To examine the structure of the file, click the View file structure... option on the Analysis menu (see View Video Stream *Structure... Ctrl+R on page 6–184).*

Opening a VOB/series of VOBs (.vob file extension)

NOTE. *This only applies to files with a VOB extension (the MTS4EA treats these files differently when compared to files with other file extensions).*

If there are two or more VOBs with sequential numbers from the first VOB selected, then MTS4EA will display a popup box:



MTS4EA Compressed Video ES Analyzer

The input file appears to be part of a series of VOB files:

V:\Temp\VTS_03_0.VOB
V:\Temp\VTS_03_1.VOB

Do you want elementary stream extraction to span these 2 files?

Clicking Yes means that the MTS4EA will extract the video Elementary Stream from all the VOBs indicated; clicking No means that the MTS4EA will only select the Elementary Stream from the VOB selected in the File open dialog box.

Once this is done, the MTS4EA then uses the standard stream selection dialog box as used for all container files (see *Opening a Container File* on page 6–19).

Opening an Uncompressed Video File (any file extension)

Uncompressed video files can be opened by selecting YUV, RGB or grayscale in the Open as type list.

In each case the MTS4EA ignores the extension and *does not* look for a compressed video standard within the data - it assumes that the data is in one of the following formats.

The YUV data is either:

- 8 bits per sample, 4:2:0
- More than 8 bits per sample, and/or 4:2:2 or 4:4:4 (as used by H.264/AVC High 10, High 4:2:2 and High 4:4:4 profiles and MPEG-2 4:2:2 profile)

YUV format of 8 bits per sample 4:2:0. (See also *YUV format of 8 bits per sample 4:2:0.*, page 6–163.) The YUV file output is raw YUV with no headers of any kind. This is the same format used by the Microsoft MPEG-4 Part 2 reference encoder Reference [7] (see *Compression Standards and File Types* section) and used commonly by other programs:

- No headers of any kind (no file or frame headers)
- One byte per sample
- Row raster order (top picture row first)
- Planar YUV 4:2:0 subsampled (4 bytes of Y data for each byte of U data and each byte of Y data)
- Y plane values are 0 - 255 unsigned
- U and V plane values are unsigned with a DC offset of 128

Other uncompressed formats. The general uncompressed video file format is as follows:

- No headers of any kind (no file or frame headers)
- Concatenated planar image data
- Row raster order (top picture row first)
- Unsigned samples

For 8-bit sample depth:

- One byte per sample

For 9-16 bit sample depth:

- Two bytes per sample
- Both little- and big-endian byte orders supported

For YUV format:

- Concatenated Y, U and V planes
- U and V planes subsampled as required
- Y plane samples are unsigned

- U and V plane samples are unsigned with a DC offset of 2^{n-1}, where *n* is the chroma sample bit depth

For RGB format:

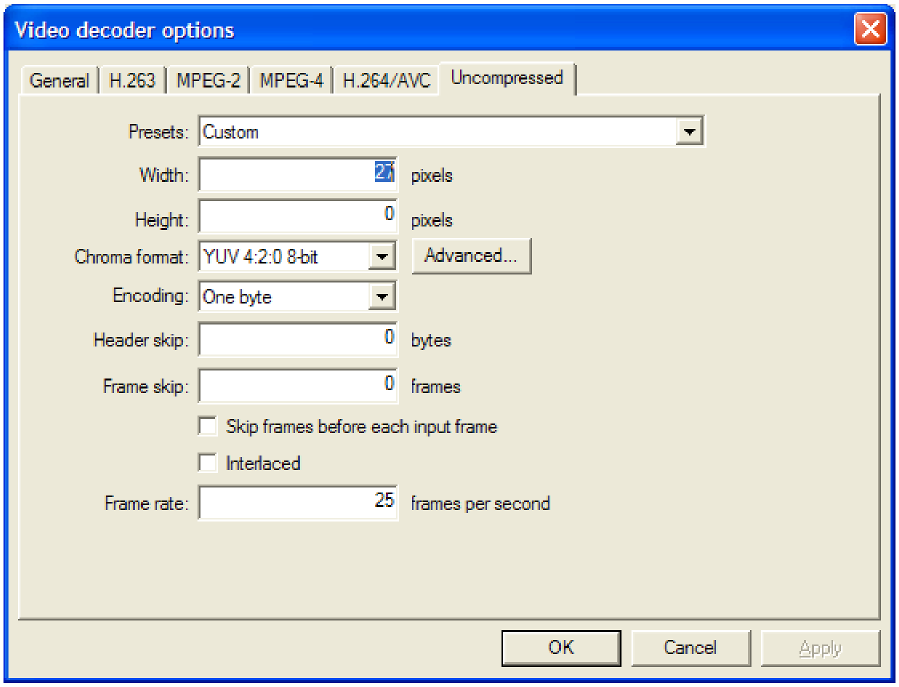
- Concatenated R, G and B planes

For grayscale format:

- Luma plane only

When opening an uncompressed video file, by default MTS4EA shows the uncompressed video decoder options, to enter the required parameters.

Setting the uncompressed video frame size, etc. Within an uncompressed video file, there is no place to indicate the frame size, frame rate, and so forth. So when the uncompressed video file is opened, the MTS4EA will display the following screen (this is a tab of the Decoder options, which can also be accessed from the Play menu):



Presets: Custom

Width: 27 pixels

Height: 0 pixels

Chroma format: YUV 4:2:0 8-bit

Encoding: One byte

Header skip: 0 bytes

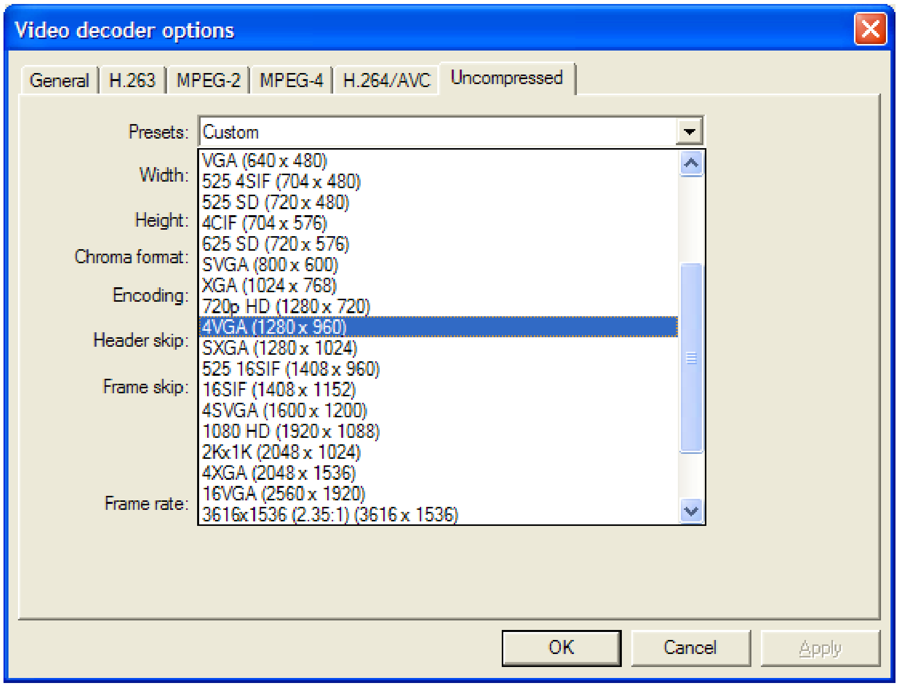
Frame skip: 0 frames

☐ Skip frames before each input frame

☐ Interlaced

Frame rate: 25 frames per second

Presets (Width and Height). If there are numbers in the filename that could indicate the frame size, then the MTS4EA will attempt to read these and offer these in the Width and Height fields shown in the previous figure. The correct values can be entered or selected using the drop-down list:



Presets: Custom

Chroma format. This control specifies the color model and format for the uncompressed video data. The combo provides several common combinations, including 8-bit YUV 4:2:0, 8-bit YUV 4:2:2, 8-bit YUV 4:4:4, 8-bit RGB and 8-bit grayscale. Select an entry from the list to use one of these common formats. Other formats can be specified by choosing the Custom entry in the list - this will open the Image format dialog.

Advanced... button. Click this button to specify a custom chroma format - this will open the Image Format dialog (see *Image format dialog* box on page 6–32).

Encoding. This control specifies the encoding of image samples in the file format. The following options are available:

- One byte - this specifies that image samples are stored in one byte per sample. This format is appropriate if all image planes are 8-bits deep
- Two byte MSB first - this specifies that image samples are stored in a pair of bytes for each sample. The most significant byte occurs first in each pair (big-endian). This format is appropriate if one or more image planes are deeper than 8-bits

- Two byte LSB first - this specifies that image samples are stored in a pair of bytes for each sample. The least significant byte occurs first in each pair (little-endian). This format is appropriate if one or more image planes are deeper than 8-bits

Header skip. The number of bytes at the start of the file before the first frame; the MTS4EA will skip past these bytes (ignoring them).

Frame skip and Skip frames before each input frame. Frame skip is the number of frames (not bytes) to skip between each frame that is viewed in the video window.

By default, these frames are skipped after each viewed frame; by selecting the Skip frames box before each input frame, the number of frames is skipped before each viewed frame.

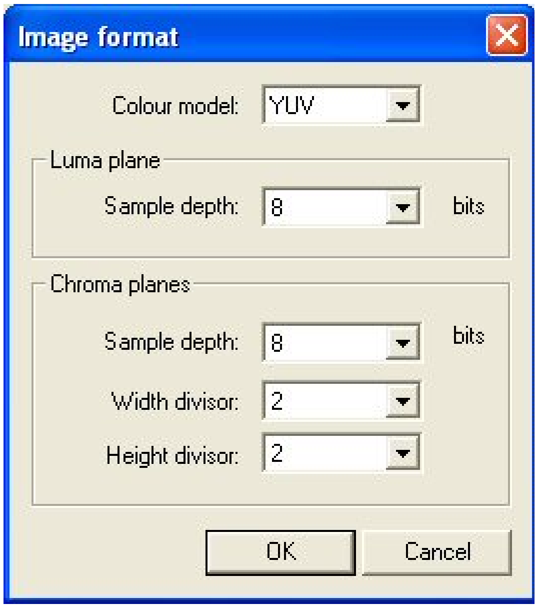
Interlaced. The uncompressed video file has interlaced data in the format of complete frames, with both fields within one frame, top-field first, each field on alternate lines.

Enabling this check box switches on the Interlace toolbar in the MTS4EA, which allows the two fields to be viewed independently, either one above the other (top field above bottom field) or either field with the field lines repeated to full frame height.

Frame rate. The rate at which to display the uncompressed frames, in frames per second.

NOTE. *The number entered in the Frame rate field can be an integer (for example, 30), or a fraction (for example, 30000/1001) or a decimal number (for example, 29.97).*

Image format dialog box. The Advanced... button opens the following window:



Colour model: YUV

Luma plane

Sample depth: 8 bits

Chroma planes

Sample depth: 8 bits

Width divisor: 2

Height divisor: 2

This dialog box allows you to specify a custom image format by choosing a color model, sample depths and sub-sampling ratios.

Color model. Use this control to choose the class of color model for the image format. Options are YUV, RGB, and grayscale.

Sample depth. The sample depth controls are used to specify the number of bits per sample for each image plane. For RGB and grayscale, only one sample depth is specified. For YUV, the luma and chroma depths can be specified separately.

Width divisor. This control specifies the horizontal sub-sampling factor for YUV chroma planes. This option is only available for YUV color models. The value is the factor by which the image width is divided to find the chroma plane width.

Height divisor. This control specifies the vertical sub-sampling factor for YUV chroma planes. This option is only available for YUV color models. The value is the factor by which the image height is divided to find the chroma plane height.

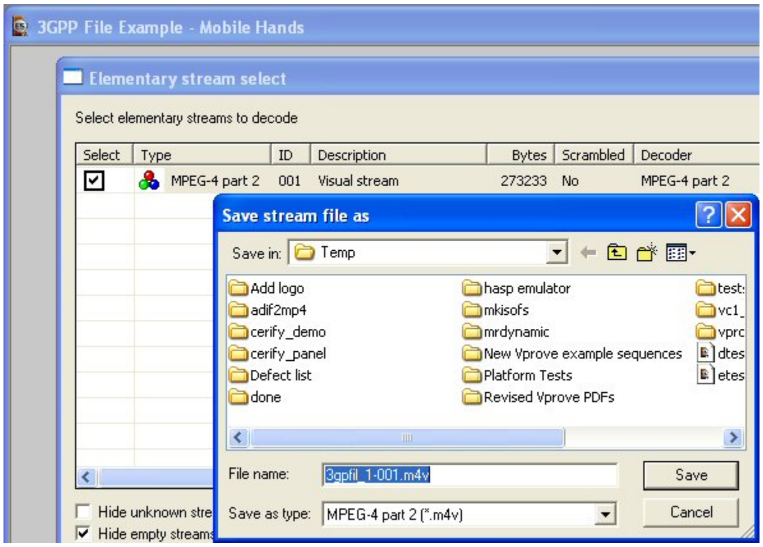
Close Stream

This allows you to close the current stream but keeps the MTS4EA window open.

Save Video Stream As...

This allows the active file to be saved to a particular file name.

This is typically used to save a video elementary stream, when this has been extracted from an MP4, 3GPP, MPG (MPEG-2), ASF or other file type, but was not saved at the time it was extracted
(see *Open Stream...* `Ctrl+O` on page 6–16):



Select elementary streams to decode

Select	Type	ID	Description	Bytes	Scrambled	Decoder
☑	MPEG-4 part 2	001	Visual stream	273233	No	MPEG-4 part 2

Save in: Temp

File name: 3gpfil_1-001.m4v

Save as type: MPEG-4 part 2 (*.m4v)

□ Hide unknown stre

☑ Hide empty streams

In this example, the video elementary stream from E-City.mp4 video track 1 is being saved to an .m4v file: the -001 (for track 1) and .m4v are automatically added by MTS4EA (because MTS4EA knows this is a 3GPP file), although any file name and extension can be entered.

The video can also be saved as an uncompressed video file, in the format given in *Opening an Uncompressed Video File (any file extension)* on page 6–27.

NOTE. *The Save stream as... feature will only save uncompressed video if the input format was also uncompressed. For compressed input, use the Decoded file output feature.*

New Project...

Some of the information saved in a project file includes the following:

- File name being analyzed (and track number, if relevant)
- List of open views and the screen location of these views
- Location of the tooltips and toolbars
- Errors that have been disabled
- Analysis options

Project Files

The project file has a file extension of `.vpp`. Double-clicking on a VPP file in Windows explorer will open the MTS4EA with the project file concerned.

The project file is actually a human-readable file in XML format. It is strongly recommended that it be not manually edited, because doing so may prevent the MTS4EA from opening a project file correctly.

Project Default File

MTS4EA has a default project file called `default.vpp`. This is used as the template when New project... is selected - all the settings in `default.vpp` are copied into a new project.

New Project

Clicking this menu item removes the current file and copies the settings from the default project file (default.vpp) to the current MTS4EA setup.

Open Project...

This allows you to open a previously saved project file.

(See also *New Project...* on page 6–34 for descriptions of projects and project files.)

Save Project... 🖫

This saves the current project to the current project file name.

(See also *New Project...* 🗋 on page 6–34 for descriptions of projects and project files.)

Save Project As...

This allows you to save the current project to a new file name.

(See also *New Project...* 🗋 on page 6–34 for descriptions of projects and project files.)

Close Project...

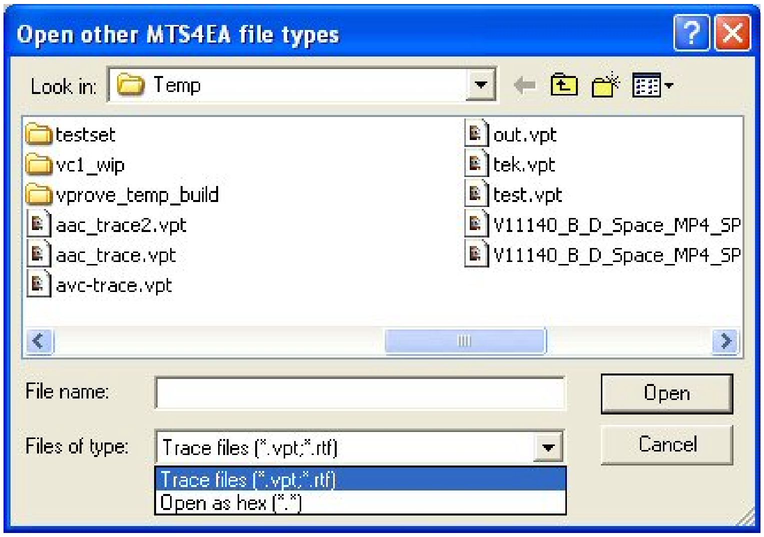
This closes the current project file.

(See also *New Project...* 🗋 on page 6–34 for descriptions of projects and project files.)

Open Other...

This allows you to browse the file system to find and open other file types, such as:

- The MTS4EA trace files, with the file extension .vpt or .rtf
- Any other file, to open in the HexView



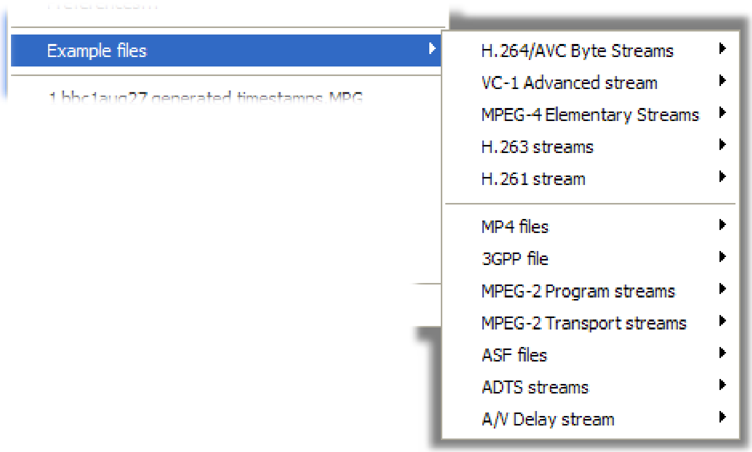
NOTE. *Opening of .rtf trace files is included for backward compatibility with older trace files generated by earlier versions of MTS4EA. From version 2.0 onwards, all trace files generated by the MTS4EA will be .vpt files. Currently, .vpt files contain only ASCII text, future versions are expected to contain additional data..*

Preferences

The Preferences dialog box allows you to select the audio playback device. Your selection will depend on the your PC setup.

Example Files...

Below this option are various example files that can be played/analyzed in the MTS4EA:



These are provided to give examples of various compressed files to experiment with and to compare with your own compressed files. Some of these are without error; others have known errors and generate appropriate Warning/Error messages (both as popup alerts and in the Trace files).

H.264/AVC Byte Streams

These are H.264/AVC compressed video Byte Streams, see *Compression Standards and File Types* section.

Six example files are provided (note that the last two are provided in MTS4EA v4.0):

Name	H.264 Profile/Level	Warnings/Errors
Neon Night	Baseline/2	- none -
Canary Wharf	Extended/3	Use of an invalid (uninitialized) Picture Parameter Set Incorrect Direct_8x8_inference_flag
Bus Junction	Baseline/3	Fails HRD conformance error (timing violation)
Grenadier Guards	Main/3	HRD buffer overflow in many frames (starting from frame 8)
Stripey Shirts	High 10	Maximum number of motion vectors per two consecutive MacroBlocks exceeded
Times Square	High 4:4:4	Maximum number of motion vectors per two consecutive MacroBlocks exceeded

NOTE. *A YUV reference file is provided for the first ten frames of the Grenadier Guards bitstream; this YUV reference file can be used for fidelity analysis and visual difference display.*

The filename is automatically filled in for the YUV reference file, but the frame rate must be set to 25.

VC-1 Advanced Profile Elementary Stream

This is a VC-1 format Advanced Profile compressed video Elementary Stream (as described in *Compression Standards and File Types* section). Advanced Profile of VC-1 is the only profile that can be a stand-alone Elementary Stream; the Simple and Main Profiles can only exist in an ASF file or in an .rcv file.

One example file is provided:

Name	VC-1 Profile	Warnings/Errors
Central Park	Advanced	Invalid value (reserved) and invalid VLC

MPEG-4 Elementary Streams

These are MPEG-4 compressed video Elementary Streams (as described in *Compression Standards and File Types* section).

Six example files are provided:

Name	MPEG-4 Profile	Warnings/Errors
Woman Drinking	Main	- none -
Train in Station	Main	- none -
Space	Simple	modulo time base Method 1 quant used
Man Walking	Advanced Simple/Level 0 (with B-VOPs, Method 1 quant)	VBV overflow
Synthetic	Advanced Simple/Level 2 (with B-VOPs, Method 1 quant, Quarter Sample)	VCV overflow
Window Car	Advanced Simple/Level 2 (with B-VOPs, Method 1 quant, GMC with 3 warping points, affine model)	stuffing bits VCV overflow

NOTE. *A YUV reference file is provided for the whole Man Walking bitstream; this YUV reference file can be used for fidelity analysis and visual difference display.*

The filename is automatically filled in for the YUV reference file, but the frame rate must be set to 30.

H.263 Streams

These are H.263 compressed video files (as described in *Compression Standards and File Types* section).

Three example files are provided:

Name	Standard/Annex	Warnings/Errors
Rally (250k)	H.263 baseline	(1) PSUPP is sent when file is H.263 baseline (non-critical warning but PSUPP field values only used in Annex L) (2) invalid variable length code for MCBPC in the last frame (also causes Out of sync error and other consequential errors)
Rally (2M)	H.263+ Annex T	PSUPP is sent but Annex L not used (non-critical warning but PSUPP field values only used in Annex L)
Person Track	H.263+ Annexes D, F, I, J, S, T	- none -

MP4 Files

These are MPEG-4 `.mp4` container files, containing compressed video and other data (as described in *Compression Standards and File Types* section).

Six example files are provided:

Name	Standard/Annex	Warnings/Errors
Packet Woman	MPEG-4 Simple Profile/Level 1	Level is set at 1; max. frame size in this Level is 176x144 pixels, but the video is 352x288 VCV overflow and VBV underflow
Piccadilly Circus	MPEG-4 Simple Profile/Level 2	VCV overflow and VBV underflow
Beijing Weather Girl	MPEG-4 Simple Profile/Level 5 (video); AAC LC (audio)	VBV overflow
Las Vegas	H.264/AVC Baseline Profile/Level 1.2	HRD information missing: no pic_timing SEI message
Live Band	MPEG-4 Simple Profile/Level 1 (video); AAC HE (audio)	Error in section data causes the calculated number of scalefactor bands to exceed the maximum allowed value
Stage Interview	MPEG-4 Simple Profile/Level 1 (video); AAC Main (audio)	None

3GPP File

This is a 3GPP `.3gp` container file, containing compressed video and other data (as described in *Compression Standards and File Types* section).

One example file is provided:

Name	Standard	Warnings/Errors
Mobile Hands	MPEG-4 Simple Profile/Level 1	Reserved value for `Profile and level indication` (set to 0)

MPEG-2 Program Streams

Three MPEG-2 `.mpg` Program Streams are provided, containing compressed video and other data (as described in *Compression Standards and File Types* section).

These examples contain audio streams, but they are all silent.

Name	Standard	Warnings/Errors
Bus Junction	MPEG-2 Main Profile/Main Level	- none -
Person Track	MPEG-2 Main Profile/Main Level	Bad slice order Invalid VLC for `dct_differential`
Grenadier Guards	MPEG-2 Main Profile/Main Level	Numerator and denominator of `frame_rate_extension` are both set to 1 but, when equal, they must be set to 0: this also generates the errors of invalid values for `frame_rate_extension_d` and `frame_rate_extension_n`

NOTE. *A YUV reference file is provided for the first ten frames of the Grenadier Guards bitstream; this YUV reference file can be used for fidelity analysis and visual difference display.*

MPEG-2 Transport Streams

Two MPEG-2 `.m2t` Program Streams are provided, containing compressed video and other data (as described in *Compression Standards and File Types* section).

Four example files are provided:

Name	Standard	Warnings/Errors
Golden Gate	H.264/AVC Main Profile	Use of undefined `disable_deblocking_filter_idc` = 3
Mangroves	MPEG-2 Main Profile/Main Level	- none -
Captain Bob	MPEG-2 Main Profile/Main Level (video); MPEG-1 Audio Layer II (audio)	- none -
British Grand Prix	MPEG-2 Main Profile/Main Level (video); MPEG-1 Audio Layer II (audio)	Error in audio header specifies an ID value and a protection bit value which do not match previous headers

ASF files

Two Microsoft ASF files are provided, containing compressed video and other data (as described in *Compression Standards and File Types* section).

Name	Standard	Warnings/Errors
Beach Girl	VC-1 Simple Profile	- none -
Great Wall	VC-1 Simple Profile	- none -

ADTS streams

Name	Standard	Warnings/Errors
Dance Track	AAC LC	Error in ADTS header specifies an incorrect syncword

A/V Delay stream

Name	Standard	Warnings/Errors
Clapperboard[1]	MPEG-2 Main Profile / Main Level (video) Dolby AC-3 (audio)	

[1] The Clapperboard file uses parts of the file archiving utility “7-Zip”.
“7-Zip is licensed under the GNU LPGL (Lesser General Project License).
See www.7-zip.org.

YUV source files for Example Fidelity Analysis

YUV source files used for encoding have been provided for example fidelity analysis, with three of the example files provided:

Standard	Stream name	YUV filename*
H.264/AVC (Main Profile/Level 3)	Grenadier Guards	guards_yuv (frames 1-9 only**: 25 frames/sec)
MPEG-2 (Main Profile/Main Level)	Grenadier Guards	guards_yuv (frames 1-9 only**: 25 frames/sec)
MPEG-4 (Advanced Simple Profile/Level 0)	Man Walking	man_walking_yuv (30 frames/sec)

NOTE. *The YUV filename is automatically filled in by MTS4EA - there is no need to enter the filename.*

For the Grenadier Guards examples, the YUV of only the first nine frames is provided, because the YUV files are so large.

Export A/V Delay Streams…

The A/V Delay measurement feature (see page 6–227) provides a complete solution for measuring audio/video synchronization delay in an encoding system. The Export A/V Delay Streams menu option allows the supplied test streams to be exported in a variety of formats for use with encoding systems.

You can use the standards Windows controls (Ctrl and Shift) to select multiple output formats for a selected stream.

NOTE. *Some of the export formats can produce very large files. Ensure that you have enough disk space to accommodate the output.*

NOTE. *The Export A/V Delay Streams feature uses parts of the file archiving utility 7-Zip. 7-Zip is licensed under the GNU LPGL (Lesser General Project License). For more information, refer to the 7-Zip Web site (www.7-zip.org).*

Both video and audio streams can be exported. You need to be careful to match the correct audio with each video file because the timings of the audio signal vary slightly depending upon the frame rate of the video that it is to be paired with.

The supplied video formats are:

- NTSC: 704x480 interlaced at 29.97… fps
- PAL: 720x576 interlaced at 25 fps
- 720p: 1280x720 progressive at 59.94… fps
- 1080i: 1920x1080 interlaced at 29.97… fps

Each video format is supplied encoded in four ways:

- H.264/AVC: High profile
- MPEG-2: Main profile
- VC-1: Advanced profile
- Uncompressed: 4:2:0 YUV, 8 bits per sample, see *YUV format of 8 bits per sample 4:2:0.*, page 6–28

The supplied audio formats are all 16-bit sampled at 48kHz:

- '25fps': to be used with video at 25 fps. This should be used with the PAL video streams.
- '29.97fps': To be used with video at 29.97… fps. This should be used with the NTSC and 1080i video streams.
- '59.94fps': To be used with video at 59.94… fps. This should be used with the 720p video streams.

Each audio format is supplied encoded in four ways:

- Dolby AC-3
- AAC
- MPEG-1 Audio Layer II
- Uncompressed: PCM 16 bits per sample at 48kHz.

Recent File

The program retains a list of the eight most recently used files, numbered 1 to 8. The recent files can be selected by highlighting them in turn.

Where a particular file could be opened in more than one view - for example, a video bitstream could be opened as a video file (in the video window) or opened in the HexView - the file is stored on the Recent File list with the last view type appended to the end:

filename.m4v [hex]	(this one is opened in the HexViewer)
filename.m4v	(this one is opened as a video file)
filename.adts	(this one is opened as an audio file)
filename.yuv [yuv]	(this one is opened as YUV video)

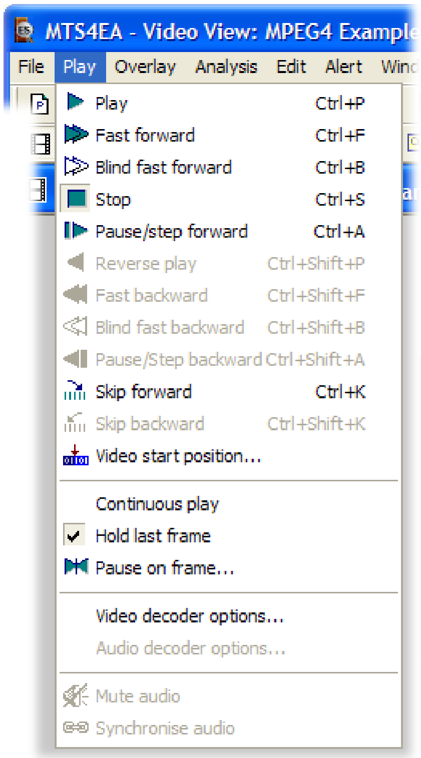
The MTS4EA then uses this information to determine in which window the file should be reopened.

Exit

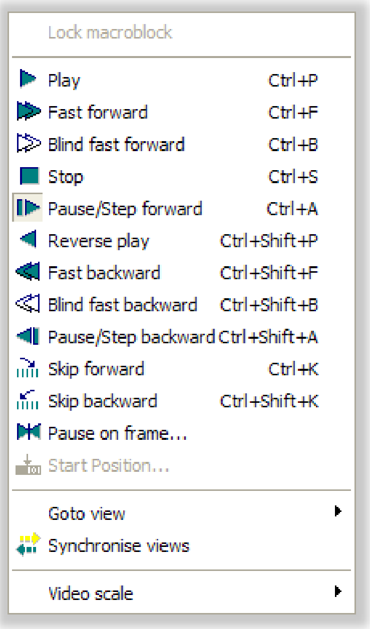
This option exits from the MTS4EA.

Play Menu

This menu controls the playing of the compressed file:


MTS4EA - Video View: MPEG4 Example
File Play Overlay Analysis Edit Alert Wind
Play Ctrl+P
Fast forward Ctrl+F
Blind fast forward Ctrl+B
Stop Ctrl+S
Pause/step forward Ctrl+A
Reverse play Ctrl+Shift+P
Fast backward Ctrl+Shift+F
Blind fast backward Ctrl+Shift+B
Pause/Step backward Ctrl+Shift+A
Skip forward Ctrl+K
Skip backward Ctrl+Shift+K
Video start position...
Continuous play
✓ Hold last frame
Pause on frame...
Video decoder options...
Audio decoder options...
Mute audio
Synchronise audio

In addition to the menu selection, there is a right-click popup Play menu:

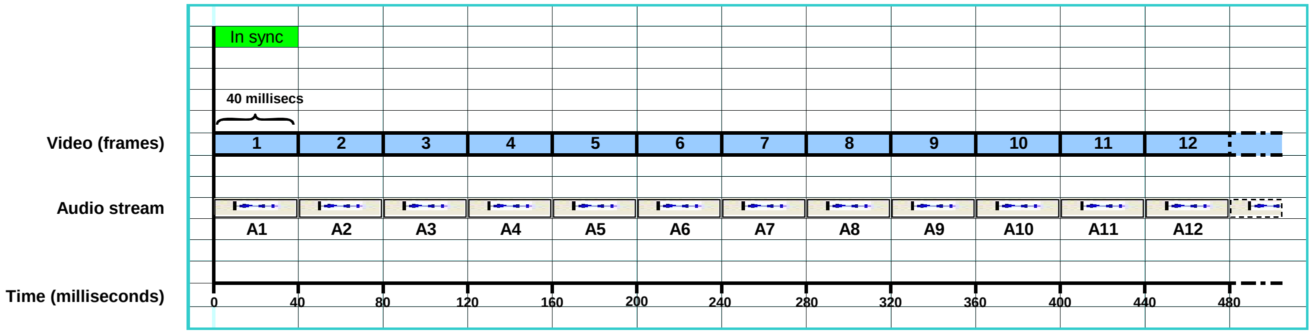

Lock macroblock
Play Ctrl+P
Fast forward Ctrl+F
Blind fast forward Ctrl+B
Stop Ctrl+S
Pause/Step forward Ctrl+A
Reverse play Ctrl+Shift+P
Fast backward Ctrl+Shift+F
Blind fast backward Ctrl+Shift+B
Pause/Step backward Ctrl+Shift+A
Skip forward Ctrl+K
Skip backward Ctrl+Shift+K
Pause on frame...
Start Position...
Goto view
Synchronise views
Video scale

The functions of each of these Play menu items are explained on the following pages. (For the items below the Play menu, see Synchronized Views/Navigating the Views on page 6–7 and Video Scale on page 6–263.)

Audio Sounds Broken Up

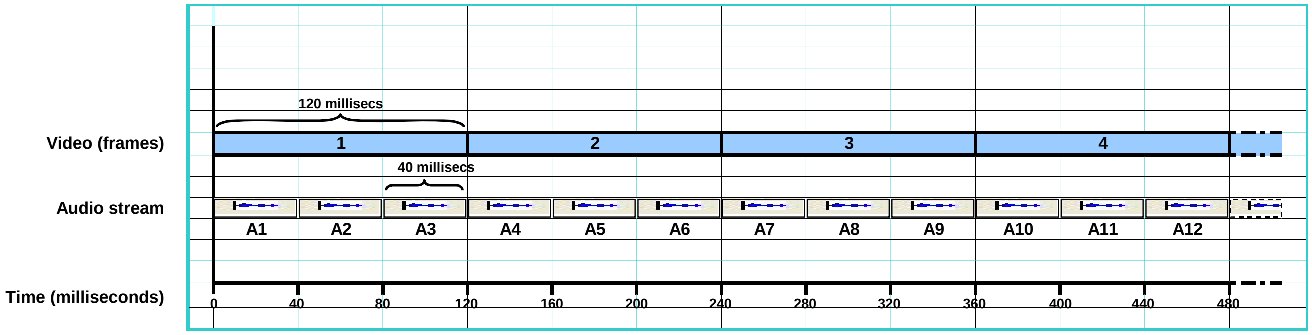
When a file containing both audio and video streams is played and the Audio synchronize function is engaged, if the PC is not sufficiently powerful, synchronization may not be possible and the resultant audio will be broken. The figures below describe, in simplified terms, how audio and video streams interrelate when decoded in the MTS4EA.

When the Synchronize audio icon is depressed, and the requested synchronization is successful, the green display appears in the Status bar and the streams run concurrently in realtime.



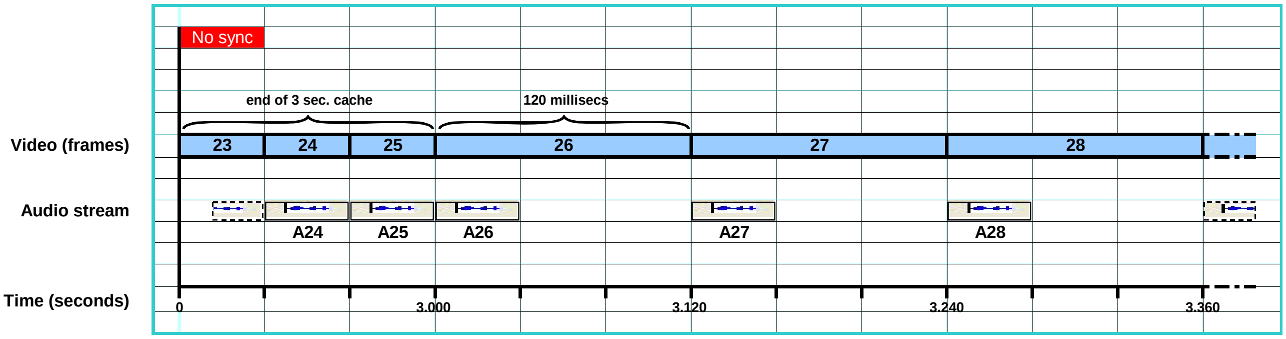
The illustration above shows video being decoded at a display rate of one frame every 40/1000 seconds. Audio plays, unbroken, at the same rate as the video.

When synchronization is not requested, the Synchronize audio icon is not depressed, and audio and video will decode independently.



The illustration above shows video playing more slowly, at a display rate of one frame every 120/1000 seconds. The audio plays at its own decode rate, outstripping the video.

When the Synchronize audio icon is depressed, and the requested synchronization is unsuccessful, the red display appears in the Status bar and the video decodes at a rate determined by available processor power, while the audio is broken.



The illustration above shows an unsuccessful attempt to synchronize audio and video: the cached data plays in sync, thereafter, limited processor power causes the audio to fragment (usually more unevenly than suggested in the above illustration) because it attempts to keep pace with the video by means of timestamps.

Audio/Video Synchronization

MTS4EA will decode and analyze audio as well as video streams, but the audio will only be played and audible when the stream is in forward play mode.

Icon	Function in video mode	Function in audio mode	Function in synchronous mode		Function in asynchronous mode	
			Video	Audio	Video	Audio
▶	Plays to end or specified frame or until paused.	Plays to end.	Attempts to play video and audio in synchrony. Whenever play is resumed, audio picks up at current location.		Audio and video play independently; no common timing enforced.	
⏩	Plays quickly in forward direction.	⏩	Video plays quickly in forward direction.	Audio muted.	Video plays quickly in forward direction.	Audio muted.
▷▷	Decodes file as quickly as possible without display.	▷▷	Decodes video file as quickly as possible without display.	Audio muted.	Decodes video file as quickly as possible without display.	Audio muted.
■	Stops.	Stops.	Stops video and audio at once.		Video stops.	Audio muted.
▮▶	Pressed once in Play mode, pauses. Pressed again, or at top of file, steps forward 1 frame.	Pause.	Pressed once in Play mode, pauses. Pressed again, or at top of file, steps forward 1 frame.	Audio muted.	Pressed once in Play mode, pauses. Pressed again, or at top of file, steps forward 1 frame.	Audio muted.
◀	Plays file backwards, at normal speed.	◀	Plays video file backwards, at normal speed.	Audio muted.	Plays video file backwards, at normal speed.	Audio muted.
⏪	Plays file quickly backwards.	⏪	Plays video file quickly backwards.	Audio muted.	Plays video file quickly backwards.	Audio muted.
◁◁	Plays file as quickly as possible backwards without display.	◁◁	Plays video file as quickly as possible backwards without display.	Audio muted.	Plays video file as quickly as possible backwards without display.	Audio muted.
◀▮	Pressed once in Play mode, pauses. Pressed again, steps back 1 frame.	◀▮	Pressed once in Play mode, pauses. Pressed again, steps back 1 frame.	Audio muted.	Pressed once in Play mode, pauses. Pressed again, steps back 1 frame.	Audio muted.
↘	Skips on to specified frame/type/time/number of frames.		Skips on to specified frame/type/time/number of frames.		Audio and video play independently; no common timing enforced.	
↖	Skips back to specified frame/type/time/number of frames.		Skips back to specified frame/type/time/number of frames.		Audio and video play independently; no common timing enforced.	
⧓	Pauses when specified video frame is reached.	⧓	Pauses when specified video frame is reached.	Audio muted.	Pauses on specified frame.	Audio muted.

Play ▶ Ctrl+P

When selected, this plays the compressed stream, video or audio, until it reaches the end of the data. However, if, when playing a compressed video stream, a frame number has been set in Pause on frame... the video will pause at this frame number.

If a file containing both audio and video streams is being played, and the streams are not synchronized, the video and audio will begin to play and continue to play independently. So, if a compressed video file is playing slowly due to the available processing power of the PC, it is likely that the video will lag behind the accompanying audio stream. If audio synchronization is selected, the rate at which the audio data is decoded adjusts and the audio stream keeps pace with the video, by means of timestamps. (See Synchronize audio on page 6–69 for more information about audio synchronization.)

NOTE. *Depending upon the speed of your PC, when playing compressed video, the standard in use and the size of the video frames being viewed, there can be a significant delay after pressing the Play or Pause/Step forward button, during which the message Buffering <nnn> (nnn is a number) is displayed near the top of the video window. This indicates that the video decoder is loading the required number of video frames into its internal buffer. This message disappears when the video is ready to display. (This is most likely to be seen with the H.264/AVC video standard, where up to 16 frames are loaded into the decoder buffer before displaying the first frame.)*

At the end of the data, the gray background screen is displayed unless:

- Hold last frame is selected, in which case the last frame stays visible
- Hold first frame is selected, in which case the first frame is re-displayed

Fast Forward ⏩ Ctrl+F

When this option is selected, the compressed file will be played in Fast forward mode, played more quickly than normal. The speed at which this is replayed depends on the speed of the computer you are using.

This command is unavailable when playing audio files.

If a file containing both audio and video streams is being played, regardless of synchronization, the video plays fast forward and the audio is muted.

NOTE. *When playing video forwards or backwards, the MTS4EA buffers the video and associated data. By default, the MTS4EA allocates a buffer size of 100 MB. The amount of buffer required per frame of video and associated data varies considerably with the video size and the video standard. Typically, 100 MB is sufficient to buffer the video and data for 10-500 frames.*

To change the buffer size, use the Play menu, Decoder options, General tab. See Video Decoder Options... on page 6–60 for more information.

The video frame count is given in a box at the top of the image, to indicate how far through the video sequence the currently displayed frame is (the frame count is also displayed at the bottom right of MTS4EA window in the status bar).

The frame count is displayed in white text if the option in the Overlay menu is currently set to White, or in black text if this is set to Black (toolbar icon ◘ or Ctrl+W).

The fast forward may be stopped at any time by clicking the Pause/Step forward icon - ❙▶ (or pressing `Ctrl+A`).

If a frame number has been set in Pause on frame... ⋈, the video will pause at this frame number.

Blind Fast Forward Ctrl+B

When this is selected, the compressed file is decoded as quickly as possible without displaying it.

If a file containing both audio and video streams is being played, regardless of synchronization, the video blind fast forwards and the audio is muted.

NOTE. *It can be useful to use Blind fast forward to quickly decode the file to see if there are any alerts generated, or to move to a selected Pause on frame… as quickly as possible.*

See the note under Fast Forward Ctrl+F *on page 6–53 regarding a possible buffering delay immediately after this is selected.*

If a frame number has been set with the Pause on frame... then the video is displayed until the designated frame number is reached.

The Blind fast forward is stopped in the same way as Fast forward.

Stop Ctrl+S

When this option is selected, it will stop the playing of the compressed file.

The gray background screen is displayed in the video window unless:

- Hold last frame is selected, in which case the last displayed frame stays visible.

- Hold first frame is selected, in which case the first frame is redisplayed.

Pause/Step Forward ▶ Ctrl+A

This option can be selected from the Play menu, by pressing Ctrl+A, or by clicking the tool bar icon Pause/Step forward ▶.

This command is available when playing audio files, but audio will only be heard when the stream is in forward play mode.

NOTE. *Repeated selection of this option will advance the displayed video one frame at a time.*

This has two functions:

- If the file is playing, selecting this option will pause the decoding of the stream and, in the case of a video file, display the most recently decoded frame
- If the file has stopped or is already paused, selecting this option causes the file to advance by one displayed frame

NOTE. *See the Note under* Fast Forward ⏩ Ctrl+F *on page 6–53 regarding a possible buffering delay immediately after this is selected.*

In H.264/AVC where SI-frames and/or SP-frames are used, when using Pause/Step forward (`Ctrl+A`), at the point of the switch, both the frames used for the switch are displayed, one after the other. These are not fully displayed in the video when it is viewed normally, and so are not displayed when using Play or Fast forward selections.

Reverse Play ◀ Ctrl+Shift+P

Play the file backwards, at normal speed.

This command is unavailable when playing audio files. Audio will only be heard when the stream is in forward play mode.

This option can be selected from the Play menu or by pressing `Ctrl+Shift+P`, or by clicking the tool bar icon Reverse play - ◀.

Fast Backward ◀◀ Ctrl+Shift+F

Play the file backwards, as fast as possible, while still displaying it.

This command is unavailable when playing audio files. Audio will only be heard when the stream is in forward play mode.

This option can be selected from the Play menu or by pressing `Ctrl+Shift+F`, or by clicking the tool bar icon Fast backward - ◀◀.

Blind Fast Backward◁◁ Ctrl+Shift+B

Play the file backwards, as fast as possible, without displaying it.

This command is unavailable when playing audio files. Audio will only be heard when the stream is in forward play mode.

This option can be selected from the Play menu or by pressing `Ctrl+Shift+B`, or by clicking the tool bar icon Blind fast backward - ◁◁.

Pause/Step Backward ◀▮ Ctrl+Shift+A

Step backwards through the video file one displayed frame at a time.

This command is unavailable when playing audio files. Audio will only be heard when the stream is in forward play mode.

See also the notes under *Pause/Step Forward* ▮▶ *`Ctrl+A`* on page 6–55.

This option can be selected from the Play menu or by pressing `Ctrl+Shift+A`, or by clicking the tool bar icon Pause/Step backward - ◀▮.

Skip Forward ↷ Ctrl+K

This will skip the video to the next frame type/time/number of frames indicated in the drop-down box on the icon toolbar.

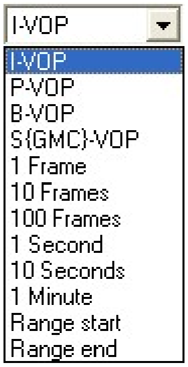
For example, if the box next to the skip forward icon has I-VOP in it (for example, ↶ ↷ I-VOP ▾) then clicking the Skip forward icon will skip to the next I-VOP.

This option can be selected from the Play menu or by pressing `Ctrl+K`, or by clicking the toolbar icon Pause/Step forward ↷.

If going to the selected frame type/time/number of frames is beyond the length of the video stream, then the stream is advanced to the end.

See also Audio/Video Synchronization on page 6–51 and Video Decoder Options... on page 6–60 regarding the buffer used to hold the video and associated data. When skipping forward or backward beyond the end of the buffer, the MTS4EA must reload the buffer, so some delay may be experienced.

The possible selections vary with the selected video standard; the example below is for MPEG-4 Advanced Simple Profile.



This command is unavailable when playing audio files. Audio will only be heard when the stream is in forward play mode.

Skip Backward Ctrl+Shift+K

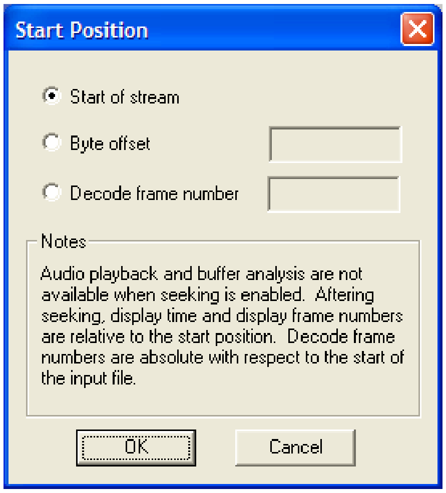
This is the same as Skip forward but in the reverse direction - see page 6–56.

This option can be selected from the Play menu, by pressing Shift+Ctrl+K, or by clicking the tool bar icon Pause/Step backward - .

This command is unavailable when playing audio files. Audio will only be heard when the stream is in forward play mode.

Video Start Position...

In the Start Position dialog box, you can set either a byte offset or decode frame number as a starting position for decoding.



Audio playback and buffer analysis are not available when seeking is enabled. Aftering seeking, display time and display frame numbers are relative to the start position. Decode frame numbers are absolute with respect to the start of the input file.

The byte offset is relative to the start of the file.

Continuous Play

This command is unavailable when playing audio files.

When this option is selected and Play/Fast forward/Blind fast forward/ Reverse play/Fast backward/Blind fast backward is clicked, then the file will be continually played in a loop. That is, the file is decoded until its end, at which point decoding will start again at the beginning of the file, until Stop or Pause is used.

Audio will only be heard when the stream is in forward play mode.

Hold Last Frame

When this option is selected and the video/audio file is played to the end, the last decoded frame will be displayed. If Hold Last Frame is not selected, the gray background video screen will be displayed when the video is not playing.

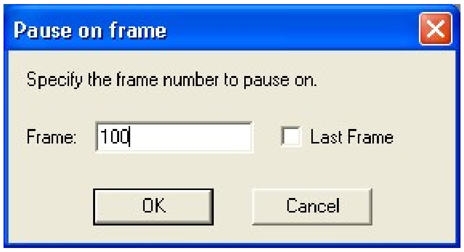
Hold Last Frame is selected as on by default when MTS4EA starts.

This command is not relevant for audio files.

Pause on Frame...

This command is only applicable to the playing of video files.

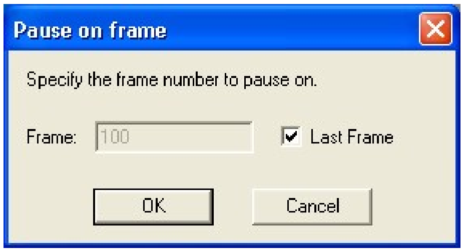
This option allows you to select a frame number to pause on when the video is played. Having selected a frame to pause on, and clicking Play or Fast forward, or Blind fast forward, the video will play until the selected frame number and then pause.


Pause on frame

Specify the frame number to pause on.

Frame: 100 □ Last Frame

The last frame can be selected by using the Last Frame box. When this is done, the video pauses on the last frame. The previously entered frame number (if there is one) is still stored, and can be enabled again by clearing the Last Frame check box:


Pause on frame

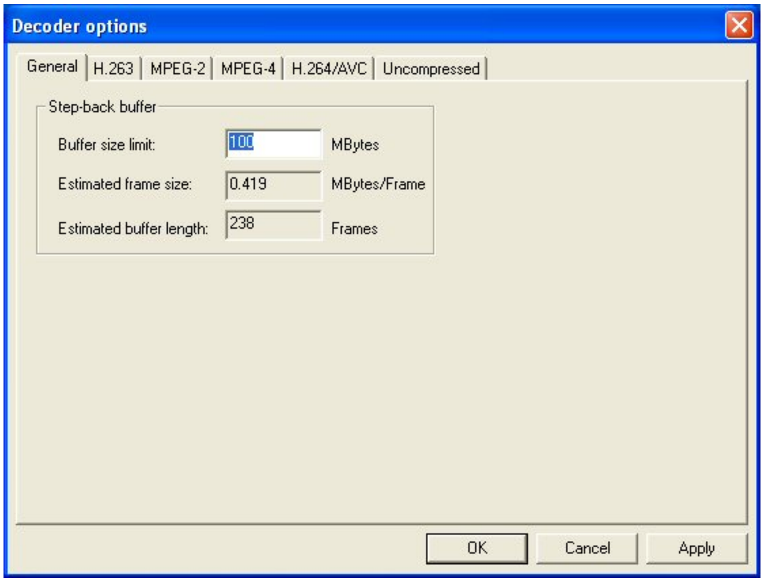
Specify the frame number to pause on.

Frame: 100 ☑ Last Frame

Video Decoder Options...

The Video Decoder dialog box has various tabs that set the video decoder options for each standard. When there are no configurable options for a particular video standard, there is no tab on the dialog.

General



Decoder options

General | H.263 | MPEG-2 | MPEG-4 | H.264/AVC | Uncompressed

Step-back buffer

Buffer size limit: 100 MBytes

Estimated frame size: 0.419 MBytes/Frame

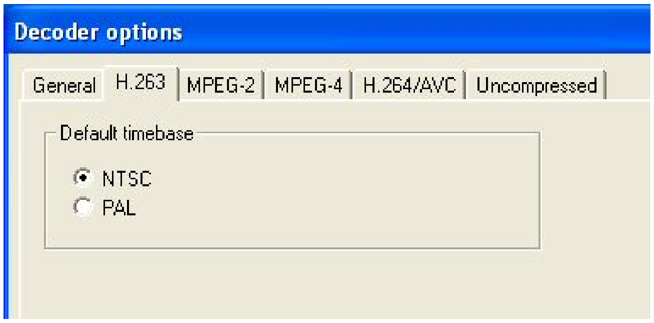
Estimated buffer length: 238 Frames

Step-back buffer. This buffer is used to hold the video and associated data, to make video play (forwards and backwards) occur without delay.

NOTE. *The size of the step-back buffer is shown, with an estimation of how many frames of data can be held within this buffer size. When playing video forwards or backwards, the MTS4EA buffers the video and associated data. By default, MTS4EA allocates a buffer size of 100 MB. The amount of buffer required per frame of video and associated data varies considerably with the video size and the video standard. Typically, 100 MB is sufficient to buffer the video and data for 10 to 500 frames.*

The popup alerts (Warnings, Errors, and so forth) that occur when video is decoded by the MTS4EA, are not displayed if the video and associated data is in the Step-back buffer – see Popup alerts in the step-back buffer. *on page 6–242 for more information.*

H.263



Default timebase

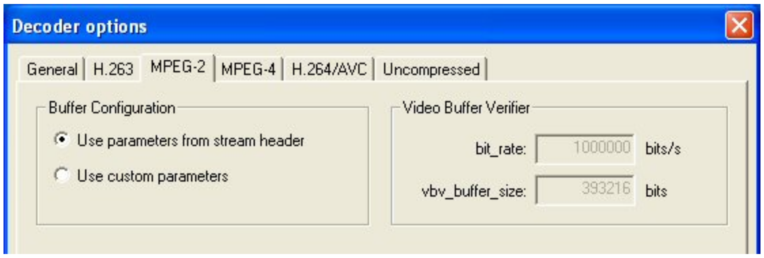
● NTSC
○ PAL

This has no effect and is grayed out for standards other than H.263.

NTSC/PAL option. Within the H.263 standard the NTSC standard is assumed (frame rate of 29.97 frames per second) and video is played back at this frame rate by default. The NTSC option is selected as standard.

However, although the video frame time can be exactly set with options in H.263+, sometimes H.263 video in PAL format (frame rate of 25 frames per second) does not set these options. In this situation PAL video would play back at 29.97 frames per second and it would play too quickly. Selecting the PAL option on this tab will play the decoded video at 25 frames per second.

MPEG-2



Buffer Configuration

● Use parameters from stream header
○ Use custom parameters

Video Buffer Verifier

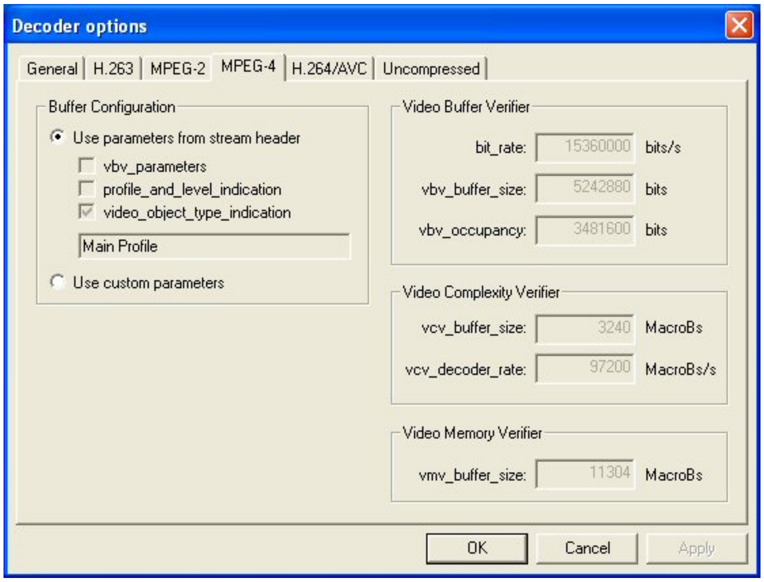
bit_rate: 1000000 bits/s
vbv_buffer_size: 393216 bits

This has no effect and is grayed out for standards other than MPEG-2.

This dialog box relates to the use of parameters for VBV buffer analysis in MPEG-2.

See *Buffer analysis controls: MPEG-4 and MPEG-2.* and *Use custom buffer parameters: MPEG-4 and MPEG-2.* on page 6–191 for more information.

MPEG-4



Buffer Configuration

● Use parameters from stream header

□ vbv_parameters

□ profile_and_level_indication

☑ video_object_type_indication

Main Profile

○ Use custom parameters

Video Buffer Verifier

bit_rate: 15360000 bits/s

vbv_buffer_size: 5242880 bits

vbv_occupancy: 3481600 bits

Video Complexity Verifier

vcv_buffer_size: 3240 MacroBs

vcv_decoder_rate: 97200 MacroBs/s

Video Memory Verifier

vmv_buffer_size: 11304 MacroBs

This has no effect and is grayed out for standards other than MPEG-4.

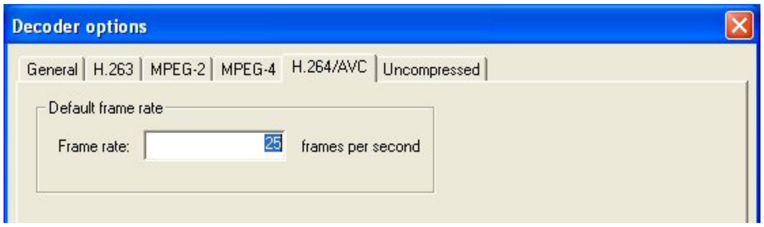
This relates to the use of parameters for buffer analysis (VBV, VCV, VMV) in MPEG-4.

See *Buffer analysis controls: MPEG-4 and MPEG-2.* and *Use custom buffer parameters: MPEG-4 and MPEG-2.* on page 6–191 for more information.

H.264/AVC

Many H.264/AVC byte streams do not provide the SEI (Supplemental enhancement information) data for specifying a frame rate, as this is optional in the byte stream (and where this happens, the frame play data is typically provided at the systems level for the video decoder).

Therefore the setting is provided on this option tab, so that the stream will play at the correct frame rate when the appropriate SEI data is not present.



Default frame rate

Frame rate: 25 frames per second

Uncompressed Video

The data is either:

- 8 bits per sample, 4:2:0, or
- More than 8 bits per sample, and/or 4:2:2 or 4:4:4 (as used by H.264/AVC High Profile/FRExt, High/10, High/4:2:2, High/4:4:4)

YUV format of 8 bits per sample 4:2:0. The YUV file output is raw YUV with no headers of any kind: this is the same format as used by the Microsoft MPEG-4 Part 2 reference encoder (see *Compression Standards and File Types* section) and as used commonly by other programs:

- No headers of any kind (no file or frame headers)
- One byte per sample
- Row raster order (top picture row first)
- Planar YUV 4:2:0 sub-sampled (4 bytes of Y data for each byte of U data and each byte of Y data)
- Y plane values are 0-255 unsigned
- U and V plane values are unsigned with a DC offset of 128

Other uncompressed formats. The general uncompressed video file format is as follows:

- No headers of any kind (no file or frame headers)
- Concatenated planar image data
- Row raster order (top picture row first)
- Unsigned samples

For 8-bit sample depth:

- One byte per sample

For 9-16 bit sample depth:

- Two bytes per sample
- Both little- and big-endian byte orders supported

For YUV format:

- Concatenated Y, U and V planes
- U and V planes sub-sampled as required
- Y plane samples are unsigned
- U and V plane samples are unsigned with a DC offset of 2^{n-1}, where *n* is the chroma sample bit depth

For RGB format:

- Concatenated R, G and B planes.

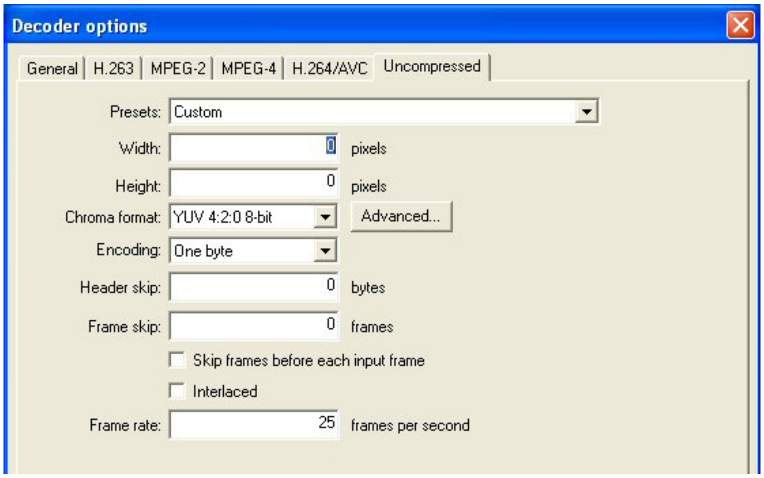
For grayscale format:

- Luma plane only

When opening an uncompressed video file, by default the MTS4EA shows the uncompressed video decoder options, to enter the required parameters.

NOTE. *If you try to open an .m4v or .264 or other compressed video file by using Open as uncompressed video, the MTS4EA will not decode the compressed video but will assume it is in uncompressed format.*

Setting the frame size. Within an uncompressed video file, there is no place to indicate the frame size, frame rate, and so forth. When the file is opened, the MTS4EA will display the following screen (this is a tab of the Decoder options, which can also be accessed from the Play menu, Decoder options):



Presets: Custom

Width: 0 pixels

Height: 0 pixels

Chroma format: YUV 4:2:0 8-bit

Encoding: One byte

Header skip: 0 bytes

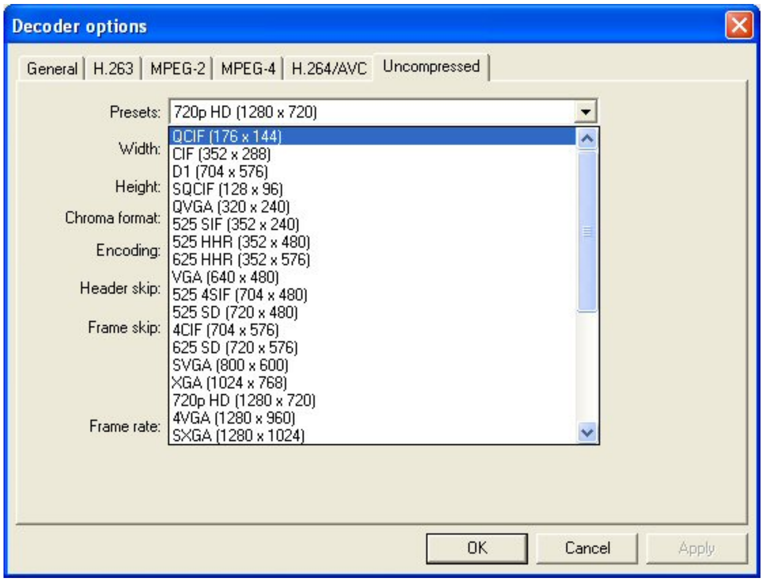
Frame skip: 0 frames

□ Skip frames before each input frame

□ Interlaced

Frame rate: 25 frames per second

Presets (Width and Height). If there are numbers in the filename that could indicate the frame size, then the MTS4EA will attempt to read these and offer these in the Width and Height fields shown in the previous screenshot. The correct values can be entered or selected using the drop-down list:



Presets: 720p HD (1280 x 720)

Chroma format. This control specifies the color model and format for the uncompressed video data. The combo provides several common combinations, including 8-bit YUV 4:2:0, 8-bit YUV 4:2:2, 8- bit YUV 4:4:4, 8-bit RGB and 8-bit grayscale. Select an entry from the list to use one of these common formats. Other formats can be specified by choosing the Custom entry in the list - this will open the Image format dialog.

Advanced... button. Click this button to specify a custom chroma format - this will open the Image Format dialog, see *Image format dialog box* on page 6–32.

Encoding. This control specifies the encoding of image samples in the file format. The following options are available:

- One byte - this specifies that image samples are stored in one byte per sample. This format is appropriate if all image planes are 8-bits deep

- Two byte MSB first - this specifies that image samples are stored in a pair of bytes for each sample. The most significant byte occurs first in each pair (big-endian). This format is appropriate if one or more image planes are deeper than 8-bits

- Two byte LSB first - this specifies that image samples are stored in a pair of bytes for each sample. The least significant byte occurs first in each pair (little-endian). This format is appropriate if one or more image planes are deeper than 8-bits

Header skip. The number of bytes at the start of the file before the first frame; MTS4EA will skip past these bytes (ignoring them).

Frame skip and Skip frames before each input frame: Frame skip is the number of frames (not bytes) to skip between each frame that is viewed in the video window.

By default, these frames are skipped after each viewed frame; by selecting the Skip frames box before each input frame, the number of frames are skipped before each viewed frame.

Interlaced: The uncompressed video file has interlaced data in the format of complete frames, with both fields within one frame, top-field first, each field on alternate lines.

Enabling this check box switches on the Interlace toolbar in the MTS4EA which allows the two fields to be viewed independently, either one above the other (top field above bottom field) or either field with the field lines repeated to full frame height.

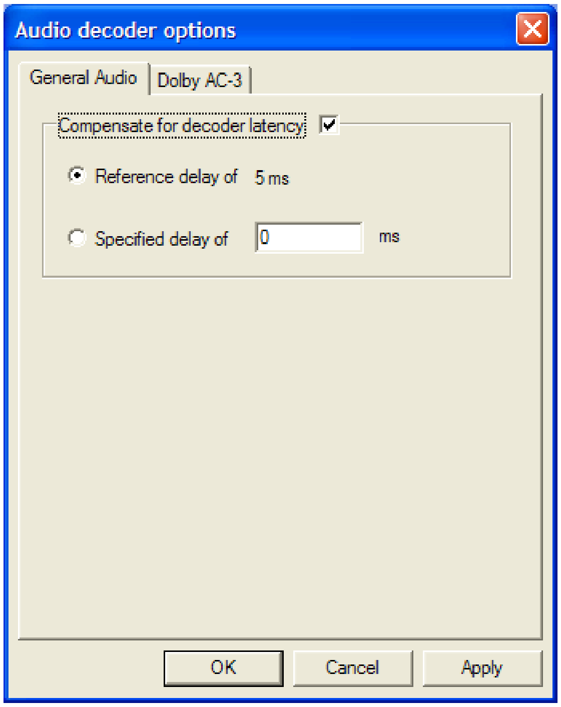
Frame rate. The rate at which to display the uncompressed frames, in frames per second.

NOTE. *The number entered in the Frame rate field can be an integer (for example, 30), or a fraction (for example, 30000/1001) or a decimal number (for example, 29.97).*

Audio Decoder Options...

The Audio Decoder option is enabled only when audio is present in a stream.

General Audio. Where a decoder has inherent latency, the MTS4EA can compensate automatically.



Audio decoder options

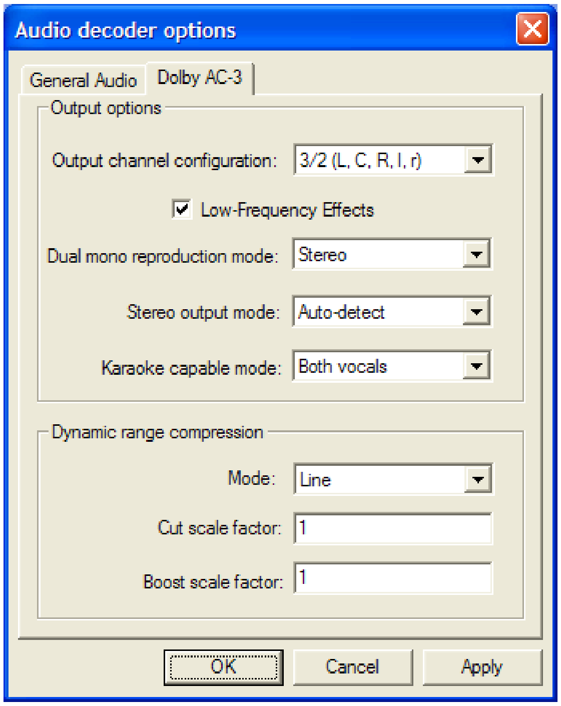
Compensate for decoder latency ☑

● Reference delay of 5 ms

○ Specified delay of 0 ms

With Compensate for decoder latency selected, a reference delay appropriate for the decoder in use is automatically available. If you want to specify a different delay, enter the value in the Specified delay field and select the option.

Dolby AC-3. The Audio Decoder Dolby AC-3 tab allows you to set up output options and dynamic range compression values.



Output options

Output channel configuration: 3/2 (L, C, R, l, r)

☑ Low-Frequency Effects

Dual mono reproduction mode: Stereo

Stereo output mode: Auto-detect

Karaoke capable mode: Both vocals

Dynamic range compression

Mode: Line

Cut scale factor: 1

Boost scale factor: 1

The fields correspond to those used in the Dolby AC-3 development kit. Changes will take effect the next time the stream is played (that is, decoded).

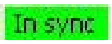
Mute audio

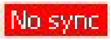
Clicking on this button turns the sound on/off. The audio stream will continue to decode, even with sound disabled.

Synchronize audio

Without synchronization enabled, audio and video streams will play independently, at rates determined by the available processing power of your PC, with audio usually playing more quickly than video. When the Synchronize audio icon is depressed, audio data will be played as the video frame to which it pertains is played. Full details can be found under *Audio/Video Synchronization* on page 6–51.

To show whether audio and video streams are successfully synchronized or not, the following displays appear in the Status bar:

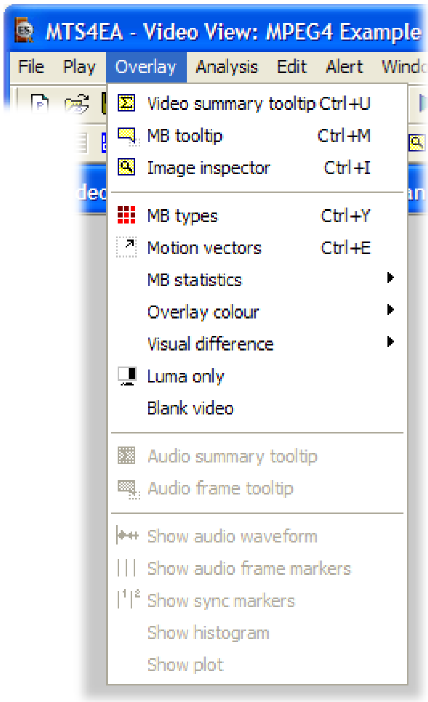

In sync


No sync

This menu item and icon are only available when a file containing both audio and video is loaded.

Overlay Menu

This menu controls the display of overlaid real-time information and statistical information that has been captured over a period of time.



This menu returns information on video streams only. The audio options for this menu are described on page 6–119.

Summary & Macroblock Tooltips Manipulation

The Summary tooltip and the MacroBlock tooltip windows move and dock in the same way.

NOTE. *When the tooltips are dragged near the edge of the window, they automatically dock. To force undocking, press and hold the Ctrl key while dragging with the mouse.*

The tooltips bring up independent windows that can be:

- Moved around within the main window and put next to a video window:

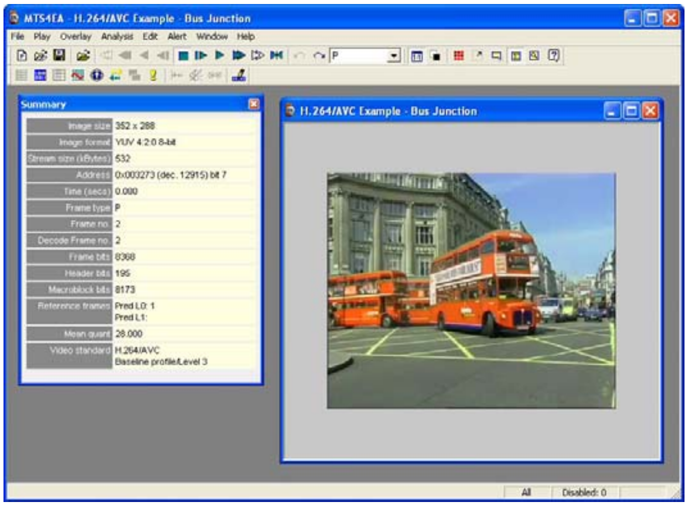

MTS4EA - H.264/AVC Example - Bus Junction
File Play Overlay Analysis Edit Alert Window Help
P
Summary
H.264/AVC Example - Bus Junction
All
Disabled: 0

Image size	352 x 288
Image format	YUV 4:2:0 8-bit
Stream size (kBytes)	532
Address	0x003273 (dec. 12915) bit 7
Time (secs)	0.080
Frame type	P
Frame no	2
Decode Frame no	2
Frame bits	8368
Header bits	195
Macroblock bits	8173
Reference frames	Pred L0: 1 Pred L1:
Mean quant	28.000
Video standard	H.264/AVC Baseline profile/Level 3

- Docked to one of the edges of the main window:

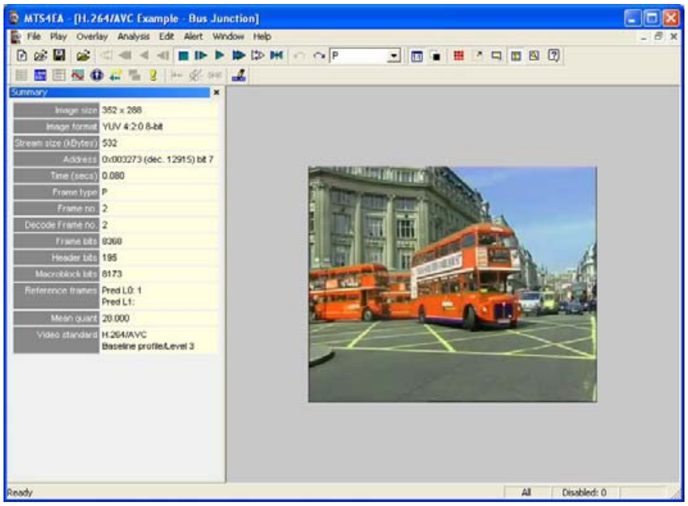

MTS4EA - [H.264/AVC Example - Bus Junction]
File Play Overlay Analysis Edit Alert Window Help
P
Summary
Ready
All
Disabled: 0

Image size	352 x 288
Image format	YUV 4:2:0 8-bit
Stream size (kBytes)	532
Address	0x003273 (dec. 12915) bit 7
Time (secs)	0.080
Frame type	P
Frame no	2
Decode Frame no	2
Frame bits	8368
Header bits	195
Macroblock bits	8173
Reference frames	Pred L0: 1 Pred L1:
Mean quant	28.000
Video standard	H.264/AVC Baseline profile/Level 3

When the Summary/MacroBlock tooltip is docked (as in the preceding figure):

- And the window is maximized, the Summary/ MacroBlock tooltip remains visible and the other window is only maximized into the space available
- The title bar of the Summary/MacroBlock tooltip changes to a pale color to indicate it is docked (and the title text is no longer bold)

Docking/undocking

- To dock the Summary/MacroBlock tooltip: drag it to one of the edges of the main window (top, left, right, bottom).

- To undock, hold the keyboard control (Ctrl) key and click on the title bar.

Scrolling/scroll bars

When the available area is too small for the whole tooltip to be seen, scroll bars are automatically displayed:

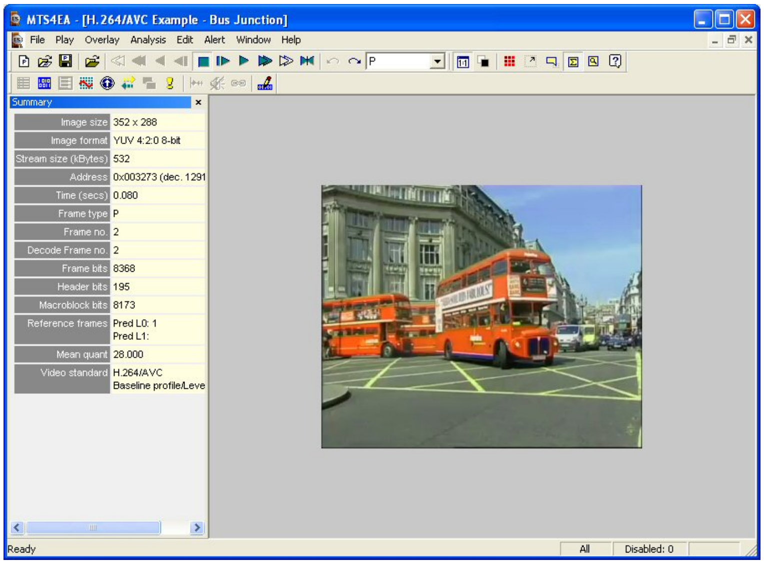


Image size	352 x 288
Image format	YUV 4:2:0 8-bit
Stream size (kBytes)	532
Address	0x003273 (dec. 1291
Time (secs)	0.080
Frame type	P
Frame no.	2
Decode Frame no.	2
Frame bits	8368
Header bits	195
Macroblock bits	8173
Reference frames	Pred L0: 1 Pred L1:
Mean quant	28.000
Video standard	H.264/AVC Baseline profile/Leve

Video Summary Tooltip Σ Ctrl+U

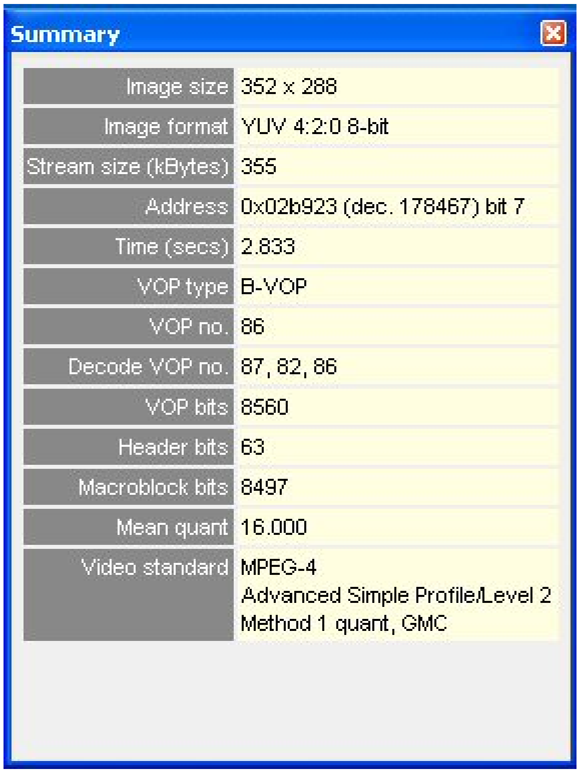


Image size	352 x 288
Image format	YUV 4:2:0 8-bit
Stream size (kBytes)	355
Address	0x02b923 (dec. 178467) bit 7
Time (secs)	2.833
VOP type	B-VOP
VOP no.	86
Decode VOP no.	87, 82, 86
VOP bits	8560
Header bits	63
Macroblock bits	8497
Mean quant	16.000
Video standard	MPEG-4 Advanced Simple Profile/Level 2 Method 1 quant, GMC

When this is switched on, a window is displayed that provides summary information about the frame being displayed or the sequence as a whole. The window can be dragged to any position within the MTS4EA main window.

NOTE. *The titles in the Summary tooltip and the displayed information vary with the selected video standard and the options that have been used in the specific video file.*

If the sequence has not been decoded to the end, then the summary up to that point is provided. The information given is described in detail below.

Tref [H.263 only]

Tref is the time to the current frame from the start of the sequence, measured in increments of the base frame time for NTSC or PAL video (~33 milliseconds or 40 milliseconds respectively). Tref has a maximum value of 255; when it reaches this value it wraps back to zero.

Skip [H.263 only]

Skip is the difference in Tref for this frame, compared to Tref for the previous frame.

NOTE. *Skip and Tref are only displayed for H.263.*

For NTSC video at roughly 30 frames per second (actually at 30,000/1,001 Hz), frames are displayed every 0.033 seconds (approximately).

However, many compressed video sequences do not have 30 frames per second of data (although they must of course be displayed at the correct rate, as if they did have 30 frames per second).

For example, if a compressed video sequence had 15 frames per second, the decoder would skip every other frame; Skip would equal two in this case. (And, for example, if NTSC video were displayed at 10 frames per second then Skip would equal three.)

For a compressed video sequence at 15 frames per second, it would mean that the displayed frame number would increase by 15 each second.

So, continuing the example of video at 15 frames per second, assuming that there are two Skips before the first frame, the values for the first few frames would be as follows:

Time approx. (secs)	Tref (Temporal reference)	Skip (increment)	Frame number (count of frames decoded and displayed
0.067	2	2	1
0.133	4	2	2
0.200	6	2	3
0.267	8	2	4

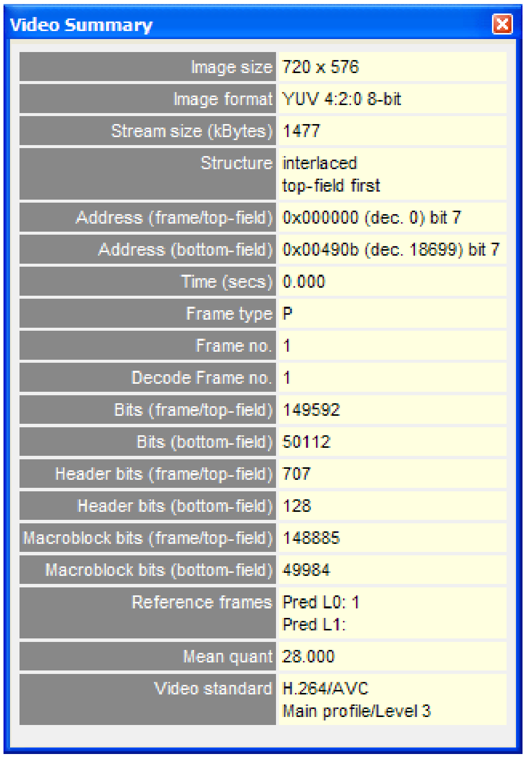
There are many variations to this; often more than one frame is skipped.

NOTE. *Also, within MTS4EA, for H.263+ and H.263 video Tref is limited to 8 bits, a maximum value of 255 as this is how Tref is defined within the H.263 standards (actually TR).*

Some of the Trace outputs of the MTS4EA keep track of the total number of frames skipped - this is called Cumul. skip (short for Cumulative skip). In the last row of the table in the preceding example Cumul. skip would be eight.

Summary Tooltip Varies by Video Standard

The information provided by the Summary tooltip varies depending upon the video standard that is being analyzed, and the specific options in the selected video file: the following is an example for an H.264/AVC stream:

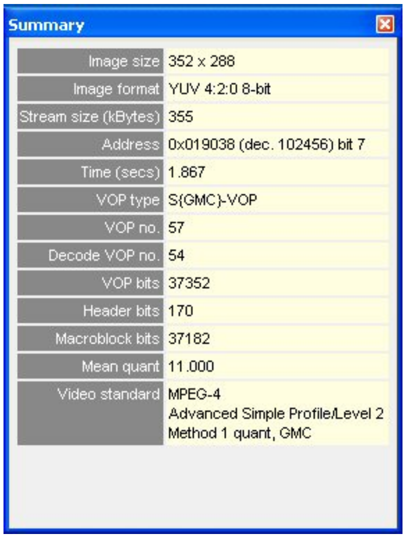


Video Summary

Image size	720 x 576
Image format	YUV 4:2:0 8-bit
Stream size (kBytes)	1477
Structure	interlaced top-field first
Address (frame/top-field)	0x000000 (dec. 0) bit 7
Address (bottom-field)	0x00490b (dec. 18699) bit 7
Time (secs)	0.000
Frame type	P
Frame no.	1
Decode Frame no.	1
Bits (frame/top-field)	149592
Bits (bottom-field)	50112
Header bits (frame/top-field)	707
Header bits (bottom-field)	128
Macroblock bits (frame/top-field)	148885
Macroblock bits (bottom-field)	49984
Reference frames	Pred L0: 1 Pred L1:
Mean quant	28.000
Video standard	H.264/AVC Main profile/Level 3

H.264/AVC Summary tooltip example

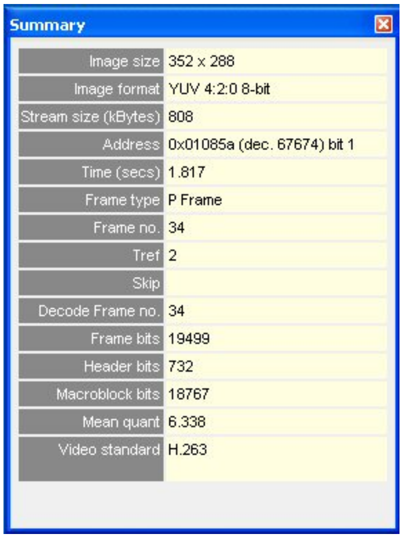
And these are examples for:



Summary

Image size	352 x 288
Image format	YUV 4:2:0 8-bit
Stream size (kBytes)	355
Address	0x019038 (dec. 102456) bit 7
Time (secs)	1.867
VOP type	S(GMC)-VOP
VOP no.	57
Decode VOP no.	54
VOP bits	37352
Header bits	170
Macroblock bits	37182
Mean quant	11.000
Video standard	MPEG-4 Advanced Simple Profile/Level 2 Method 1 quant, GMC

MPEG-4 Advanced Simple Profile



Summary

Image size	352 x 288
Image format	YUV 4:2:0 8-bit
Stream size (kBytes)	808
Address	0x01085a (dec. 67674) bit 1
Time (secs)	1.817
Frame type	P Frame
Frame no.	34
Tref	2
Skip	
Decode Frame no.	34
Frame bits	19499
Header bits	732
Macroblock bits	18767
Mean quant	6.338
Video standard	H.263

H.263

NOTE. *The information displayed at the end of video sequence is different – see under Summary Tooltip at the end of a Video Sequence on page 6–83.*

Image size [All standards]. Picture size in pixels, Width x Height (pixel Columns x Rows).

For H.264/AVC where the cropping rectangle feature has been used, the full image size is given and displayed in the video window, but the cropped image size is also given in this field of the tooltip in the form:

Image size	352x288	(cropped: 200x100)

Image format [All standards]. Format of selected video file

Stream size (kBytes) [All standards]. Size of the video file in KB.

NOTE. *For container files such as MP4, 3GPP and MPG (MPEG-2), the file size given is the size of the video track that has been selected, not the size of the whole MP4/3GPP/MPG file.*

Structure [H.264/AVC, VC-1 Adv. Profile and MPEG-4 ASP]. For H.264/AVC byte streams and MPEG-4 Advanced Simple Profile, where interlace could be used, the structure of the displayed frame is one of the following:

- Progressive
- Interlaced

Address or Address (frame/top-field) and Address (bottom-field) [Interlace]. The start address of the current video frame/VOP within the video file.

The H.264/AVC byte streams, two address fields are given to display the address of both fields in interlaced streams. If a stream is not interlaced, then the Address (bottom field) has no data.

NOTE. *The first byte in the video file is byte 0; the first bit is bit 7 of byte 0.*

The byte address is displayed in hexadecimal first, for example, 0x002681, then in decimal in brackets, for example, (dec. 9857) and finally the bit location within the byte, where bit 7 is the most significant bit (occurs first in the stream).

Time (secs) [All standards]. The time in seconds from the start to that point in the sequence, calculated as if the sequence was playing normally (MTS4EA keeps track of the playing time, regardless of whether the sequence is paused at some point or played in fast forward mode).

This information is calculated from:

- [H.264/AVC]: the SEI parameters if they are present in the bitstream.

NOTE. *Many H.264/AVC bitstreams do not have the relevant SEI values (these are optional) and in this case, MTS4EA will play the video at the frame rate indicated in the H.264/AVC tab of the Decoder options (on the Play menu)*

- [VC-1]: the parameters either in the ASF file, if the video is within this, or within the VC-1 stream
- [MPEG-4]: the parameters such as `modulo_time_base`, `vop_time_increment`, etc.
- [MPEG-2]: the parameters `frame_rate_value`, `frame_rate_extension_n` and `frame_rate_extension_d`
- [H.263]: the time-stamps in the picture header. If the Temporal references [Tref] are not set or are not correctly set, this time may not be the actual playing time from the start of the sequence.

VOP type [MPEG-4] or Frame type [H.264/AVC, MPEG-2, H.263, VC-1].

NOTE. *For interlaced bitstreams, the frame/VOP type reported is that relating to the second field in a frame. This means that if the top field is first in frame/VOP 1 (and this will always be Intra-coded) and the bottom field is Inter [P] coded, then the frame will be reported as P.*

- [MPEG-4] the VOP type, I-VOP or P-VOP or B-VOP or S-GMC VOP (see the following):
  - I-VOP: indicates that this frame is Intra coded (completely coded within itself) without any interpolation from earlier or later frames. The first frame in an MPEG-4 sequence is always an I-VOP
  - P-VOP: P or Predicted VOP: this frame is Inter coded (partly coded based upon earlier frames)
  - B-VOP: *(MPEG-4 Advanced Simple Profile only)* B or Bidirectional Interpolated: this frame is calculated based upon both earlier and later frames. B-VOPs may only be interpolated based upon I-VOPs/P-VOPs (not on other B-VOPs)
  - S-GMC: *(MPEG-4 Advanced Simple Profile only)* the VOP is an S-GMC VOP which uses Global Motion Compensation (GMC)
- [H.264/AVC, VC-1] the Frame type, I-frame or P-frame or B-frame or SI-frame or SP-frame (see below):
  - I-frame: I indicates that this frame is Intra coded (completely coded within itself) without any interpolation from earlier or later frames. The first frame in an H.264/AVC sequence is always an I frame
  - P-frame: P indicates Predicted: this frame is Inter coded (partly coded based upon earlier frames)
  - B-frame: *(Extended Profile only)* B or Bidirectional Interpolated: this frame is calculated based upon both earlier and later frames
  - SI-frame: SI indicates that this frame is a Switching-Intra coded frame - there is a switch between two different streams at this point
  - SP-frame: SP indicates that this frame is a Switching-Inter coded frame - there is a switch between two different streams at this point

NOTE. *For H.264/AVC, the frame type is derived from the most complex slice type present. For example, if any of the slices in the frame are B-slices, the whole frame is reported as a B-frame. If any of the slices in the frame are P-slices, the whole frame is reported as a P-frame. (This is as suggested in the standard.)*

- [MPEG-2] the Frame type, I-frame or P-frame or B-frame (see the following):
  - I-frame: I indicates that this frame is Intra coded (completely coded within itself) without any interpolation from earlier or later frames. The first frame in an MPEG-2 Elementary Stream sequence is always an I frame
  - P-frame: P indicates Predicted: this frame is Inter coded (partly coded based upon earlier frames)
  - B-frame: B or Bidirectional Interpolated: this frame is calculated based upon both earlier and later frames
- [H.263] Frame type: the frame type, I-frame or P-frame or B-frame (see the following):
  - I-frame: I indicates that this frame is Intra coded (completely coded within itself) without any interpolation from earlier or later frames. The first frame in an H.263 sequence is always an I frame
  - P-frame: P indicates Predicted: this frame is Inter coded (partly coded based upon earlier frames)

VOP no. [MPEG-4] or Frame no. [H.264/AVC, MPEG-2, H.263, VC-1]: . The number of the displayed VOP [MPEG-4]/frame [H.264/AVC, MPEG-2, H.263] in the sequence.

NOTE. *For sequences with B-VOPs/B-frames and in some other cases, the displayed frame number may be different from the decode for the VOPs/frames - see the following.*

Tref and Skip [H.263 only]: See under *Tref [H.263 only]* on page 6–74.

Decode VOP no. [MPEG-4] or Decode Frame no. [H.264/AVC, MPEG-2, H.263, VC-1]. The data that appears in this field differs from the video standard.

For H.264/AVC, one number is displayed (the current display frame number) as each MacroBlock can be bi-directionally predicted from different frames; to find out which frames are used for prediction, the MacroBlock tooltip is used.

For MPEG-4 Advanced Simple Profile and MPEG-2, when B-VOPs/B-frames are used, the data in these VOPs/frames is calculated from data in preceding and following VOPs/frames stored in the video file.

That is, for sequences with B-VOPs the order in which the VOPs/frames are stored in the video file is different from the order in which the VOPs/frames are displayed.

In this circumstance, this field of the Summary tooltip shows the source VOPs/frames in the video file from which the displayed VOPs/frames are generated.

The three numbers, xxx, bbb, fff, are:

- xxx is the frame number of the decoded VOP
- bbb is the frame number of the backward reference
- fff is the frame number of the forward reference

NOTE. *When B-VOPs are used, it means that the display is different from the decode order in every VOP after the first B-VOP occurs in the bitstream (in the display order).*

VOP bits [MPEG-4] or Bits (frame/top-field) and Bits (bottom-field) [H.264/AVC, VC-1] or Frame bits [H.264/AVC, MPEG-2, H.263, VC-1]. The number of bits used in that:

- [MPEG-4] VOP
- [H.264/AVC, VC-1] frame if progressive coded or top-field if interlaced; data only appears in the Bits (bottom-field) if the frame is interlaced.
- [MPEG-2, H.263, VC-1] frame

NOTE. *For H.264/AVC, if any of the slices in the frame are interlaced, the whole frame is reported as interlaced.*

Header bits [All standards] or Header bits (frame/top-field) and Header bits (bottom-field) [Interlace]. The sum of non-MacroBlock data bits within the frame - , the header bits at the start of a frame that are before the bits specific to the first MacroBlock in the frame.

MacroBlock bits [All standards] or MacroBlock bits (frame/top-field) and MacroBlock bits (bottom-field) [Interlace]. The sum of MacroBlock data bits within the frame - those bits specifically used to encode the MacroBlocks (excluding the header bits at the start of the frame).

PSNR (Y, U, V) [All standards] or PSNR (Y, U, V) (frame/top-field) and PSNR (Y, U, V) (bottom field) [Interlace]. When fidelity analysis is enabled, the selected fidelity analysis values are displayed for the VOP/frame currently being displayed.

The Y, U, V values given are the average of the values for each MacroBlock.

Mean quant [All standards]. The mean average of all coded MacroBlock quant values for this frame.

Reference frames [H.264/AVC only]. This shows the List 0 and List 1 reference frames to which the current display frame refers.

NOTE. *The images of the reference frames can easily be seen using the Video navigator view.*

Video standard [All standards]. The video standard used, with various additional information as appropriate, such as:

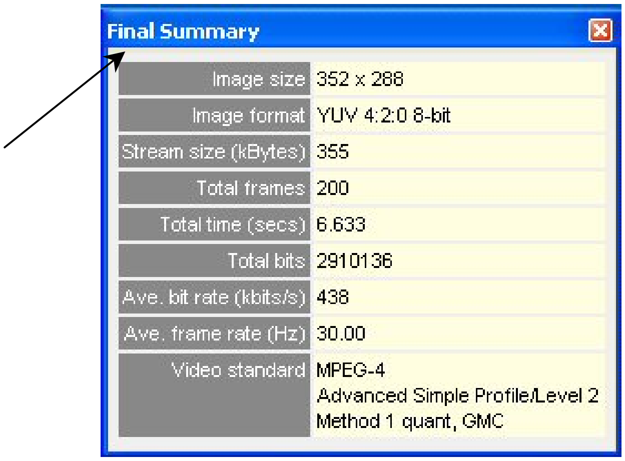
- [MPEG-4]:
  - Profile and Level, if this information is given in the file
  - Data partitioning, Resync, RVLC, as appropriate
  - [MPEG-4 Advanced Simple Profile] Interlace, QS, Method 1 quant, as appropriate
- [MPEG-2, VC-1]:
  - Profile and Level
- [H.264/AVC]:
  - Profile and Level
  - Entropy coding mode, CAVLC

NOTE. *For H.264/AVC, many of the other elements (for example, data partitioning) can vary by frame/slice (and in some cases MacroBlock by MacroBlock), so these other elements are reported in the MacroBlock tooltip.*

- [H.263] Annexes, as appropriate.

Summary Tooltip at the end of a Video Sequence

When the file is completely decoded, the summary tooltip displays as follows:



The title of the tool tip becomes Final Summary

Final Summary

Image size	352 x 288
Image format	YUV 4:2:0 8-bit
Stream size (kBytes)	355
Total frames	200
Total time (secs)	6.633
Total bits	2910136
Ave. bit rate (kbits/s)	438
Ave. frame rate (Hz)	30.00
Video standard	MPEG-4 Advanced Simple Profile/Level 2 Method 1 quant, GMC

The meanings of Image size, File size (KB) and Standard are the same as described in *Video Summary Tooltip* Σ Ctrl+U on page 6–74.

The other items displayed are:

Total frames. The total decoded frames in the sequence.

Total time (secs). The total time in seconds from the start to the end of the sequence, calculated as if the sequence were playing normally (MTS4EA keeps track of the playing time, regardless of whether the sequence is paused at some point or played in fast forward mode).

Total bits. The size of the video file in bits.

NOTE. *For container files such as MP4 and 3GPP, the size given is the number of bits in the video track that has been selected (not the container file).*

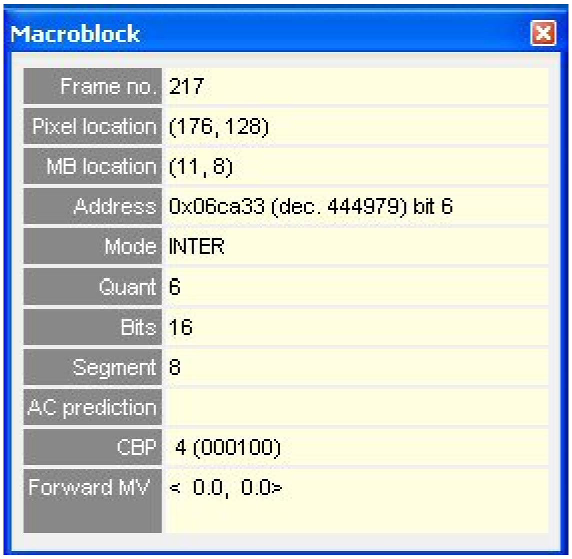
Ave. bit rate (kbits/s). This is Total bits divided by Total time, in k bits per second.

Ave. frame rate (Hz). This is Total frames divided by Total time, in frames/second (Hz).

Ave. PSNR (Y, U, V) [All standards] or Ave. PSNR (Y, U, V) (frame/top-field) and Ave. PSNR (Y, U, V) (bottom field) [Interlace]. When fidelity analysis is enabled, the average of the selected fidelity analysis values across all the VOPs/frames is displayed.

MacroBlock Tooltip Ctrl+M

This displays information about a selected MacroBlock:



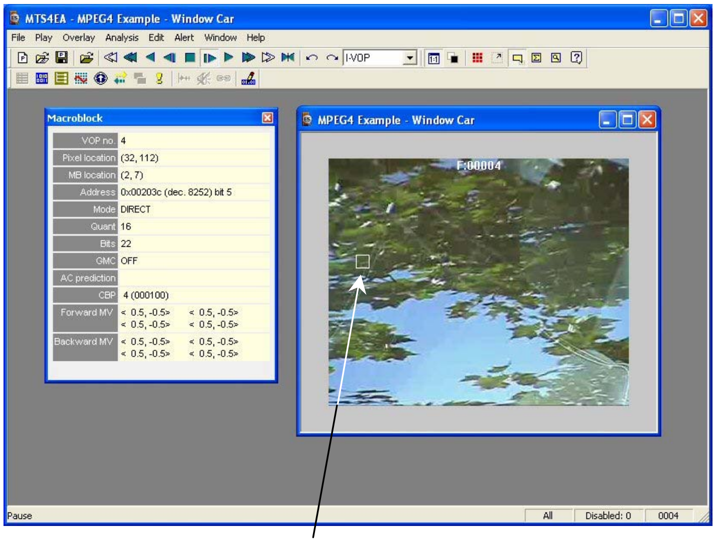
Macroblock

Frame no.	217
Pixel location	(176, 128)
MB location	(11, 8)
Address	0x06ca33 (dec. 444979) bit 6
Mode	INTER
Quant	6
Bits	16
Segment	8
AC prediction	
CBP	4 (000100)
Forward MV	< 0.0, 0.0>

NOTE. *The information displayed by the MB tooltip varies with the video standard and the options used within the particular video sequence (see under Viewing the MacroBlock data in other Views on page 6–85).*

The example above is for an MPEG-4 sequence, and is examining an Inter MacroBlock.

The MacroBlock selected is signified by a white delineation box around the 16x16 MacroBlock border:


MTS4EA - MPEG4 Example - Window Car
File Play Overlay Analysis Edit Alert Window Help
I-VOP
Macroblock
MPEG4 Example - Window Car
F:00004
Pause
All
Disabled: 0
0004

VOP no.	4
Pixel location	(32, 112)
MB location	(2, 7)
Address	0x00203c (dec. 8252) bit 5
Mode	DIRECT
Quant	16
Bits	22
GMC	OFF
AC prediction	
CBP	4 (000100)
Forward MV	< 0.5, -0.5> < 0.5, -0.5> < 0.5, -0.5> < 0.5, -0.5>
Backward MV	< 0.5, -0.5> < 0.5, -0.5> < 0.5, -0.5> < 0.5, -0.5>

Move the hand cursor to select the MacroBlock

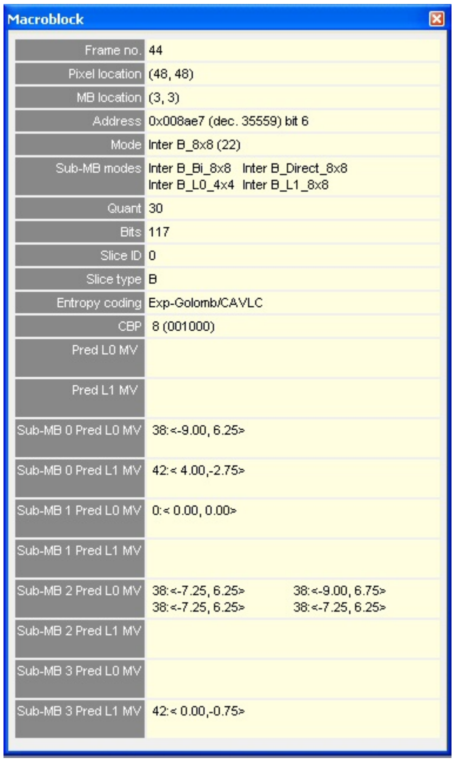
Viewing the MacroBlock data in other Views

To see the MacroBlock data in other views - for example, the HexView - right click at the MacroBlock location of interest and select Goto view from the context menu, then select one of the views. For example, if Hex View is selected, the first byte of the selected MacroBlock is displayed in the HexView.

See *Synchronized Views/Navigating the Views* on page 6–7 for more information.

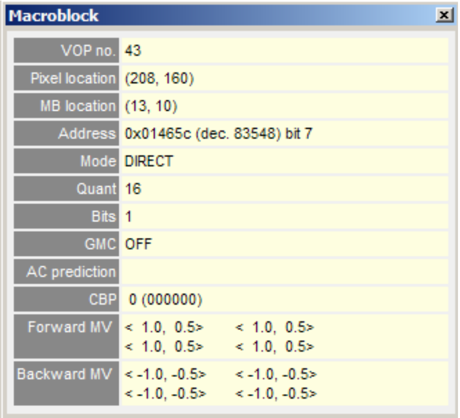
MacroBlock tooltip by Video Standard

The information provided by the Macroblock tooltip varies depending upon the video standard that is being analyzed, and the options used in the video file:



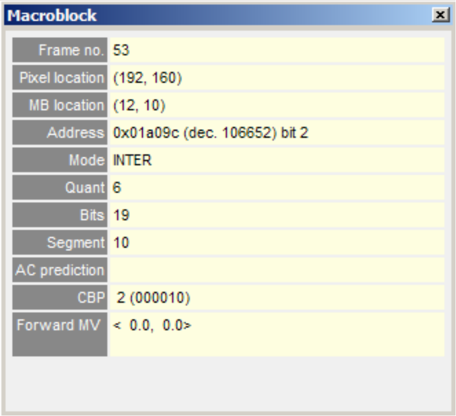
Frame no.	44
Pixel location	(48, 48)
MB location	(3, 3)
Address	0x008ae7 (dec. 35559) bit 6
Mode	Inter B_8x8 (22)
Sub-MB modes	Inter B_Bi_8x8 Inter B_Direct_8x8 Inter B_L0_4x4 Inter B_L1_8x8
Quant	30
Bits	117
Slice ID	0
Slice type	B
Entropy coding	Exp-Golomb/CAVLC
CBP	8 (001000)
Pred L0 MV	
Pred L1 MV	
Sub-MB 0 Pred L0 MV	38:<-9.00, 6.25>
Sub-MB 0 Pred L1 MV	42:< 4.00,-2.75>
Sub-MB 1 Pred L0 MV	0:< 0.00, 0.00>
Sub-MB 1 Pred L1 MV	
Sub-MB 2 Pred L0 MV	38:<-7.25, 6.25> 38:<-9.00, 6.75> 38:<-7.25, 6.25> 38:<-7.25, 6.25>
Sub-MB 2 Pred L1 MV	
Sub-MB 3 Pred L0 MV	
Sub-MB 3 Pred L1 MV	42:< 0.00,-0.75>

H.264/AVC



VOP no.	43
Pixel location	(208, 160)
MB location	(13, 10)
Address	0x01465c (dec. 83548) bit 7
Mode	DIRECT
Quant	16
Bits	1
GMC	OFF
AC prediction	
CBP	0 (000000)
Forward MV	< 1.0, 0.5> < 1.0, 0.5> < 1.0, 0.5> < 1.0, 0.5>
Backward MV	< -1.0, -0.5> < -1.0, -0.5> < -1.0, -0.5> < -1.0, -0.5>

MPEG-4 Intra MB
(in a Data partitioned VOP)



Frame no.	53
Pixel location	(192, 160)
MB location	(12, 10)
Address	0x01a09c (dec. 106652) bit 2
Mode	INTER
Quant	6
Bits	19
Segment	10
AC prediction	
CBP	2 (000010)
Forward MV	< 0.0, 0.0>

H.263 Inter MB (note Segment)

Picture type [Interlace only]. Shows whether the MacroBlock belongs to a top- or bottom-field.

VOP no. [MPEG-4] or Frame no. [H.264/AVC, VC-1, MPEG-2, H.263]. The number of the displayed VOP [MPEG-4]/frame [H.264/AVC, MPEG-2, H.263] in the sequence.

In MPEG-4 Advanced Simple Profile, H.264/AVC and MPEG-2 for sequences with B-VOPs/B-frames, the *displayed* VOP/frame number is different from the currently *decoded* VOP/frame number for every VOP/frame after the first B-VOP/B-frame occurs in the bitstream (in the *display* order) - see *Video Summary Tooltip* Σ Ctrl+U on page 6–74 for more information on decoded versus displayed VOP/frame numbers.

Pixel location [All standards]. The location of the top-left corner pixel of the MacroBlock, where:

- 0, 0 is the top-left corner of the displayed image
- The first digit is horizontal (X) location and the second digit is the vertical (Y) location.

MB location [All standards]. The location of the MacroBlock where:

- 0, 0 is the top-left MacroBlock of the displayed image
- The first digit is horizontal (X) count and the second digit is the vertical (Y) count of MacroBlocks. For example, for a CIF-size image this will go from 0,0 (top left MacroBlock) to 21,17 (bottom right MacroBlock).

Address [All standards]. The start address of the MacroBlock within the video file.

NOTE. *The first byte in the video file is byte 0; the first bit is bit 7 of byte 0.*

For container files such as MP4, 3GPP and MPG (MPEG-2), the address given is the offset from the start of video track that has been selected, not the address within the container file (which will be different).

The byte address is displayed in hexadecimal first, for example, 0x002b8a then in decimal in brackets, for example, (dec. 11146) and finally the bit location within the byte, where bit 7 is the most significant bit (occurs first in the stream).

Frame/Field coding [H.264/AVC, VC-1, MPEG-4 Advanced Simple Profile and MPEG-2]. This displays the MacroBlock coding as either frame (progressive) or field (interlaced).

2nd DP part, 3rd DP part [optional, MPEG-4 only]. Within the MPEG-4 standard, the video data within a VOP can be divided into sections for error resilience purposes, by using the Data Partitioned flag. In this situation, the data for each MacroBlock is located at three different areas in the bitstream.

The Address field displays the address of the first part of the MacroBlock data. The 2nd DP part and 3rd DP part display the addresses of the second and third parts of the Data Partitioned MacroBlock data respectively.

Mode [All standards]. The MacroBlock type/coding mode as given in *MacroBlock Types* ▦ *Ctrl+Y* on page 6–94.

Sub-MB modes [H.264/AVC and VC-1 Advanced Profile only]. The type/coding mode of the blocks within the MacroBlock as given in H.264/AVC on page 6–95.

Quant [All standards]. The Quantizer used by the Luminance (from 1 to 31 for all standards except H.264/AVC which is the range 0 to 51).

Bits [All standards]. The total number of bits used to encode the MacroBlock (including motion vector data). For Data partitioned MacroBlocks, this is the number of bits used in all three parts of the Data partitioned data for the MacroBlock.

GMC [optional, MPEG-4 Advanced Simple Profile only]. This denotes whether the MacroBlock uses Global Motion Compensation (GMC) in the calculation of motion vectors. This is either:

- ON – GMC is on
- OFF – GMC is off

Segment [H.263 only]. The segment number, representing the GOB number or slice (starting at zero from the top of the frame).

Slice [MPEG-2 only]. The number of the slice (starting at zero from the top of the frame).

Slice ID [H.264/AVC and MPEG-4 VC-1 Advanced Profile only]. The number of the slice (starting at zero from the top of the frame).

Slice type [H.264/AVC and MPEG-4 VC-1 Advanced Profile only]. The slice type, I or P or [Extended Profile only] B or SI or SP.

Entropy coding [H.264/AVC and MPEG-4 VC-1 Advanced Profile only]. The entropy coding mode of the slice, CAVLC.

Transform [MPEG-4 Advanced Simple Profile and VC-1 Advanced Profile]. Shows the DCT type: field or frame, as determined by the value of dct_type in MPEG-4 ASP interlaced_information().

AC prediction [MPEG-4, H.264/AVC, and H.263]. This is used differently for MPEG-4 and for H.263:

- MPEG-4:
  - ON – AC prediction is on
  - OFF – AC prediction is off
- H.263, Advanced Intra Coding mode as specified in Annex I, the values used are:
  - 0 – DC prediction only
  - 1 – Vertical DC & AC prediction
  - 2 – Horizontal DC & AC prediction

CBP [All standards]. Coded Block Pattern: this is a number from 0-63, signifying which of the six 8x8 blocks that make up a MacroBlock have been coded - the bit pattern is reproduced for the block in brackets, where reading left to right shows which of the six blocks were coded (1 = coded, 0 = not coded). The blocks are ordered as Y0, Y1, Y2, Y3, U, V.

Forward MV [MPEG-4, MPEG-2, and H.263 only] and Forward MV (frame/top-field), Forward MV (bottom field) [Interlace]. The Motion Vectors for the MacroBlock. The numbers in <> brackets indicate the relative position of the area from where the data for this MacroBlock has come. The numbers are for X and Y respectively (positive numbers are down/to the right; negative numbers are up/to the left).

For example, <-7.5, -5.0> means that the data in this MacroBlock came from the area of the preceding VOP/frame, which is 7.5 pixels up and 5.0 pixels to the left of the current MacroBlock.

(The term Forward in this field is used to specify that these Motion Vectors are forward predicted – they have been predicted forward from a preceding VOP/frame.)

[MPEG-4 Advanced Simple Profile] For interlaced MacroBlocks, this shows which fields the forward Motion Vectors predict from, as determined by the value of forward_top_field_reference and forward_bottom_field_reference in MPEG-4 Adv. Simple Profile interlaced_information(). This is displayed as follows:

Display	Definition
Forward motion vector (frame/top-field)	In frame coded MacroBlocks, shows the forward motion vectors. In field coded MacroBlocks, shows the forward top-field motion vectors.
Forward motion vector (bottom-field)	In field coded MacroBlocks, shows the forward bottom-field motion vectors.

Backward MV [MPEG-4 Advanced Simple Profile (B-VOPs), MPEG-2 and VC-1 Advanced Profile] and Backward MV (frame/top-field), Backward MV (bottom field) [Interlace]. These only appear for B-VOPs in MPEG-4 Advanced Simple Profile, MPEG-2 B-frames and VC-1 Advanced Profile; these are the Backward predicted Motion Vectors. The numbering follows the convention given above for Forward motion vectors.

(The term Backward in this field is used to specify that these Motion Vectors are backward predicted, they have been predicted backward from a following VOP/frame.)

[MPEG-4 Advanced Simple Profile] For interlaced MacroBlocks, this shows which fields the forward Backward Vectors predict from, as determined by the value of backward_top_field_reference and backward_bottom_field_reference in MPEG-4 Adv. Simple Profile interlaced_information(). This is displayed as follows:

Display	Definition
Backward motion vector (frame/top-field)	In frame coded MacroBlocks, shows the backward motion vectors. In field coded MacroBlocks, shows the backward top-field motion vector.
Backward motion vector (bottom-field)	In field coded MacroBlocks, shows the backward bottom-field motion vector.

Pred L0 MV, Pred L1 MV [H.264/AVC, MPEG-4 and VC-1 Advanced Profile]. The Motion Vectors for the MacroBlock. Data only appears in this field if there is no subdivision of the MacroBlock:

- L0 refers to List 0 reference frames (forward reference, from past frames; as an example, P type MacroBlocks can only be List 0)
- L1 refers to List 1 reference frames (is forward and backward reference, from past and future frames; as an example, B type MacroBlocks can be List 1 or List 0).

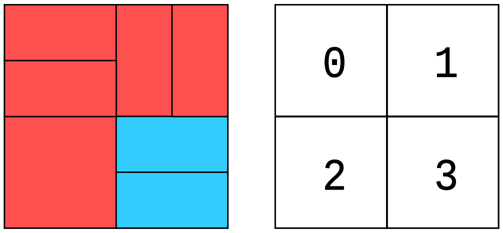
For H.264/AVC only, the number of the referenced frame is also included in front of the Motion Vector, for example, 38:<9.00,6.25> means that these motion vectors reference frame 38.

NOTE. *The images of the reference frames can easily be seen using the Video navigator view.*

Sub-MB 0/1/2/3 Pred L0/L1 MV [H.264/AVC, VC-1 Advanced Profile and MPEG-4]. The Motion Vectors for the blocks within the MacroBlock, when the MacroBlock has been subdivided (when not subdivided, there is no data in these fields).

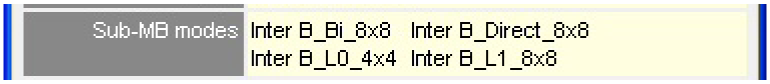
The data provided here follows the standard and is shown in the layout as given in the Sub-MB modes field of the MacroBlock tooltip and MB types overlay.

As an example, for a MacroBlock which has been divided as follows:



where the red blocks are `Inter_List_0` (P) and the blue blocks are `Inter_List_1` (B) and the numbers denote the sub-MacroBlock areas 0, 1, 2, 3.

Then the Sub-MB modes field would look like this:

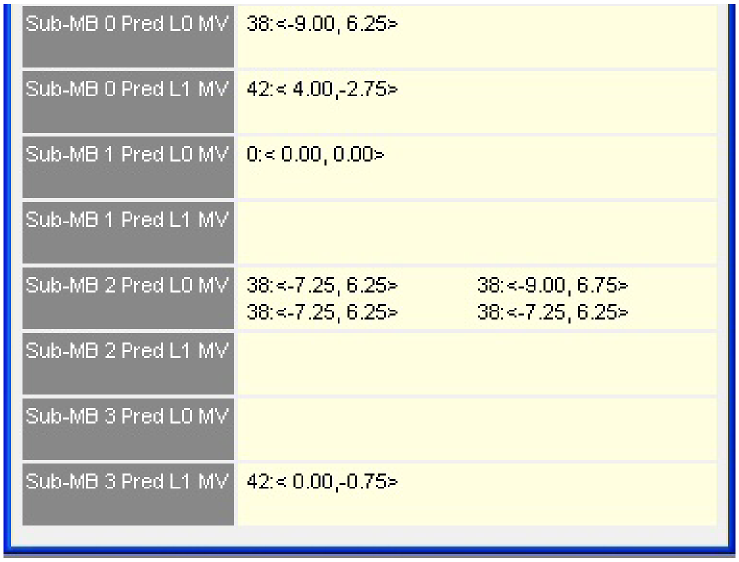


Sub-MB modes	Inter B_Bi_8x8	Inter B_Direct_8x8
	Inter B_L0_4x4	Inter B_L1_8x8

where:

- The top-left reported sub-MB mode is for area 0
- The top-right is for area 1, and so forth.

The MV fields would look like this:



Sub-MB 0 Pred L0 MV	38:<-9.00, 6.25>	
Sub-MB 0 Pred L1 MV	42:< 4.00,-2.75>	
Sub-MB 1 Pred L0 MV	0:< 0.00, 0.00>	
Sub-MB 1 Pred L1 MV		
Sub-MB 2 Pred L0 MV	38:<-7.25, 6.25> 38:<-7.25, 6.25>	38:<-9.00, 6.75> 38:<-7.25, 6.25>
Sub-MB 2 Pred L1 MV		
Sub-MB 3 Pred L0 MV		
Sub-MB 3 Pred L1 MV	42:< 0.00,-0.75>	

PSNR (Y, U, V) or **PSNR (Y, U, V) (frame/top-field)** and **PSNR (Y, U, V) (bottom field) [Interlace]**. When fidelity analysis is enabled, the selected fidelity analysis value for the selected MacroBlock is displayed.

Image Inspector Ctrl+I

The Image inspector provides a magnifying glass on the decoded video and the ability to view pixel data for individual Macroblocks separated into the component channels (YUV, RGB or grayscale).

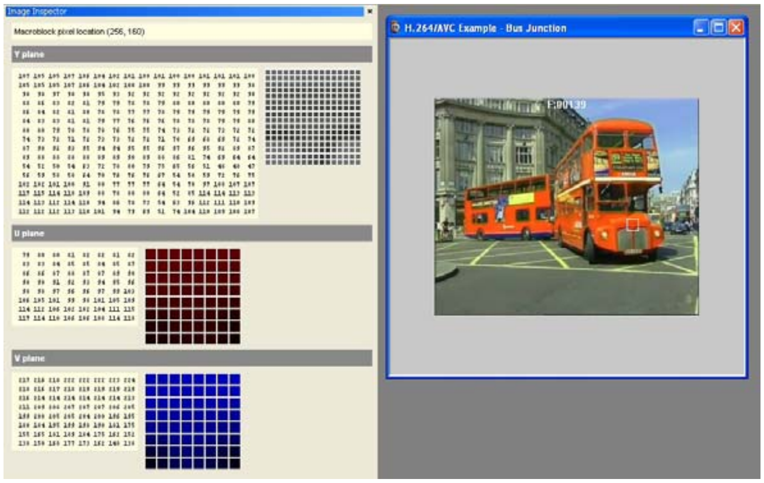
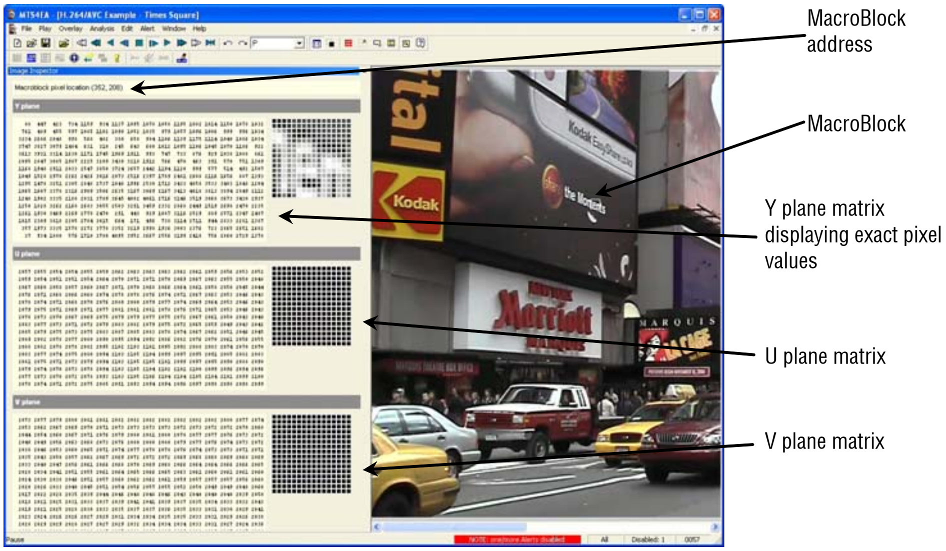


Image Inspector
Macroblock pixel location (256, 160)
Y plane
U plane
V plane
H.264/AVC Example - Bus Junction
F:00139

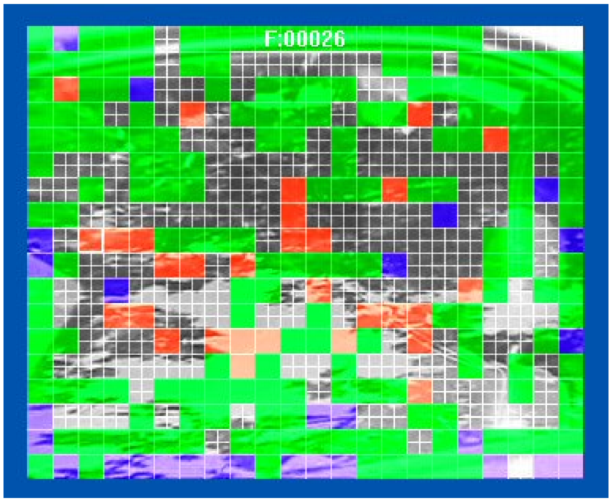
The subsampling ratio of these component channels can vary. As can be seen in the preceding screenshot, when this is the case, the matrices containing the pixel data are sized accordingly. The preceding screenshot shows an example of 8 bits per sample, 4:2:0 YUV video, the following screenshot shows an example of 12 bits per sample, 4:4:4 video.


MTS4EA - [H.264/AVC Example - Times Square]
File Play Overlay Analysis Edit Alert Window Help
Image Inspector
Macroblock pixel location (352, 208)
Y plane
U plane
V plane
MacroBlock address
MacroBlock
Y plane matrix displaying exact pixel values
U plane matrix
V plane matrix

The colors of the image blocks change, depending on the data type:

	Top image block	Middle image block	Bottom image block
YUV	Gray (Y)	Red (Cr)	Blue (Cb)
RGB	Red	Green	Blue

MacroBlock Types ▦ Ctrl+Y



The example above is a B-VOP of an MPEG-4 Advanced Simple Profile stream.

This overlay allows you to easily view the MacroBlock types used in the encoding. The types are identified by changing the color of the individual MacroBlocks, by modifying the chrominance while leaving the luminance unchanged, so that the data is still partially visible.

The colors used are dependent on the video compression standard in use. The colors are shown in the following tables and are also displayed on-screen in the MacroBlock types color key tooltip (see *MacroBlock Types Color Key Tooltip* on page 6–277 for more information).

The lines denote the edges of the MacroBlock, and if the MacroBlock has four motion vectors, the lines also divide the MacrBlock into four quadrents. The color of the lines can be changed from white to black by clicking the ◘ icon (or clicking the Black menu item on the Overlay menu).

H.264/AVC

H.264/AVC has the following possible coded MacroBlock types (see Reference [13] in *Compression Standards and File Types* section), shown in the following colors:

MB type	Prediction mode	Table index	Sub-MB types	Color	Slice type(s)
I	Intra 16x16 (incl. Intra inferred)	1 - 25	-	Green (Green)	I, P, B, SI, SP
I	Intra 4x4	0	-	Yellow (Yellow)	I, P, B, SI, SP
I	Intra 8x8	0	-	Orange (Orange)	I, P, B, SI, SP
P	Inter list 0	0 - 2	0 - 3	Red (Red)	P, SP
B	Inter list 0	1, 4, 5	1, 4, 5, 10	Red (Red)	B
B	Inter list 1	2, 6, 7	2, 6, 7, 11	Blue (Blue)	B
B	Inter list 0 + 1	3, 20, 21	3, 8, 9, 12	Pink (Pink)	B
B	Inter mixed	8 - 19	-	Light blue (Light blue)	B
B	Inter direct	0	-	Gray (Gray)	B
SI	Intra 4X4 and 16x16	0	-	Green (Green)	SI
SP	Inter list 0	0 - 2	0 - 3	Red (Red)	SP

NOTE. *The Intra inferred MacroBlock type was shown in yellow in previous versions of MTS4EA; however, because it occurs infrequently and it is a 16x16 type, it is now grouped in the Intra 16x16 MacroBlock type, and the Intra 4x4 MacroBlock type is now shown in yellow.*

VC-1

VC-1 has the following possible coded MacroBlock types (see Reference [17] in *Compression Standards and File Types* section), shown in the following colors:

Type	Motion vector	Color	Frame type(s)
Intra	0	Green (Green)	I, P, B, BI
Forward	0, 1, 2, or 4	Red (Red)	P, B
Backward	0, 1, 2, or 4	Blue (Blue)	B
Bi-directional	0, 2, or 4	Pink (Pink)	B

MPEG-4

MPEG-4 has the following possible coded MacroBlock types (see Reference [1] - Table B-1 in *Compression Standards and File Types* section), shown in the following colors:

Index	Type	Dquant/ Dbquant	Motion vector	Color	VOP type(s)
0	Inter		1	Red (Red)	P, S-GMC
1	Inter+Q	✓	1	Blue (Blue)	P, S-GMC
2	Inter4V		4	Pink (Pink)	P, S-GMC
3	Intra			Green (Green)	I, P, S-GMC
4	Intra+Q	✓		Yellow (Yellow)	I, P, S-GMC
-	Direct		8	Gray (Gray)	B
-	Interpolate	✓	2	Green (Green)	B
-	Forward	✓	1	Red (Red)	B
-	Backward	✓	1	Blue (Blue)	B

MPEG-2

MPEG-2 has the following possible coded MacroBlock types (see Reference [14] in *Compression Standards and File Types* section), shown in the following colors:

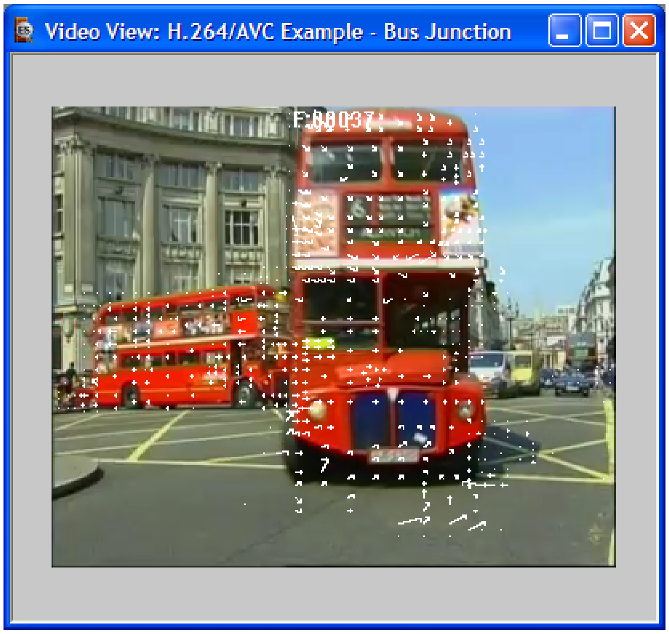
Type	Dquant	Motion vector	Color	Frame type(s)
Intra	✓	1	Green (Green)	I, P, B
Forward	✓	2	Red (Red)	P, B
Backward	✓	2	Blue (Blue)	B
Bi-directional	✓	4	Pink (Pink)	B

H.263+ and H.263

H263 has six possible coded MacroBlock types (see Reference [2] - Table 9/H.263 - in *Compression Standards and File Types* section), shown in the following colors:

Index	Type	Dquant	Motion vector	Color
0	Inter		1	Red (Red)
1	Inter+Q	✓	1	Blue (Blue)
2	Inter4V		4	Pink (Pink)
3	Intra			Green (Green)
4	Intra+Q	✓		Yellow (Yellow)
5	Inter4V+Q	✓	4	Gray (Gray)

Motion Vectors ↗ Ctrl+E

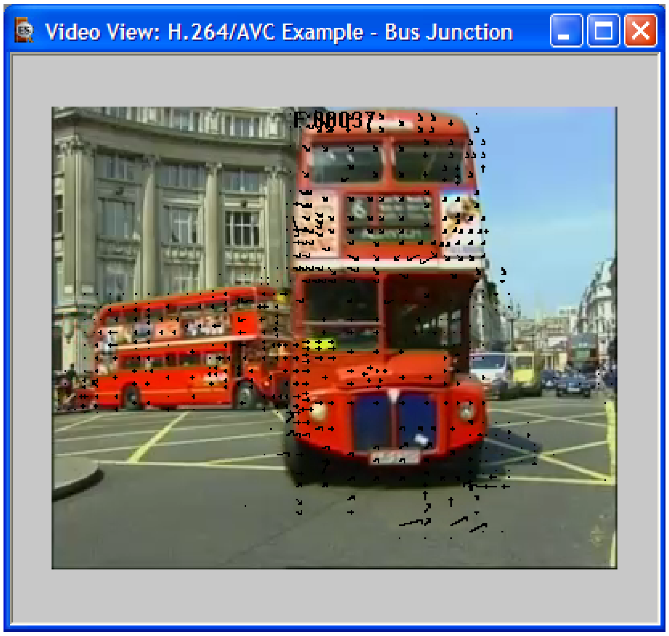


The motion vectors used in relevant video frames can be displayed on the frame currently being viewed. They are switched on and off by doing one of the following:

- From the Overlay menu, select the Motion vectors option
- Click the toolbar icon ↗
- Press `Ctrl+E`

The motion vectors are drawn in the image in either white or black (the color can be selected using the Black/White Digits button on the Toolbar ■ see *Overlay Color* ■ *Ctrl+W* on page 6–113) for all frames except B-VOPs/B-frames - see **Motion** Vectors in B-VOPs/B-frames on page 6–100 for information on this.

This is an example of the above video frame with the motion vectors plotted in black:



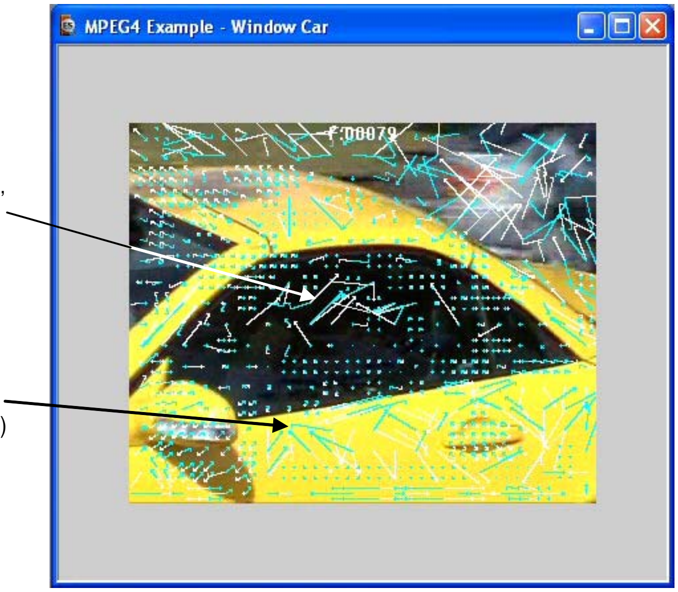
The motion vectors are plotted to the length of that actually used by the decoder.

The vector has an arrowhead at its end and is pointing to the center of the region of pixels in the previous frame that was used for the prediction of the current MacroBlock or sub-MacroBlock.

Some video standards allow motion vectors for an 8x8 region; other video standards (such as H.264/AVC) allow motion vectors for smaller regions as well. In these situations, all the motion vectors are plotted.

Motion Vectors in B-VOPs/B-frames

For frames that are bidirectionally predicted (B-VOPs or B-frames) there can be two motion vectors for each MacroBlock or sub-MacroBlock region, or four MacroBlocks in the case of interlaced video sequences:



Forward motion vector, displayed in white (or black)

Backward motion vector, displayed in light blue (or dark blue)

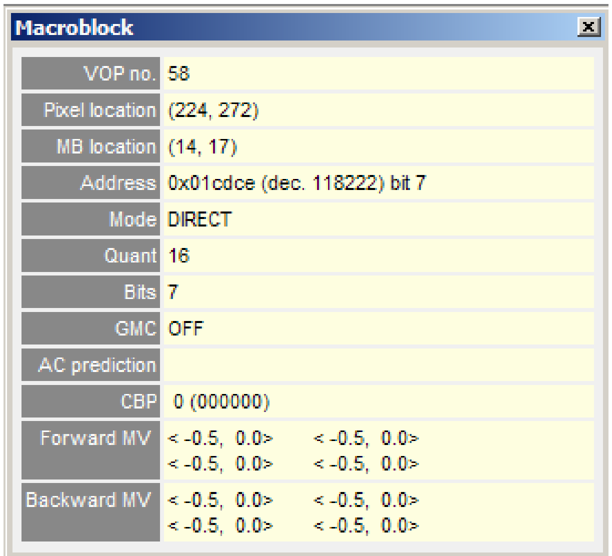
In Interlaced video sequences the motion vectors are displayed in the following colors:

MV Type	Field	MV color - ⧉ out	MV color - ⧉ out
Forward	Top	(White)	Black (Black)
Backward	Top	Light blue (Light blue)	Dark blue (Dark blue)
Forward	Bottom	Yellow (Yellow)	Green (Green)
Backward	Bottom	Magenta (Magenta)	Red (Red)

For H.264/AVC, the white (or black) arrows denote the List 0 motion vectors and the light blue (or dark blue) arrows denote the List 1 motion vectors.

For more information on the motion vector displays in the H.264/AVC standard, see *Compression Standards and File Types* section.

For MacroBlocks with four (or more) motion vectors, the MacroBlock tooltip is expanded with extra information to display all the motion vectors. The example below is for a MacroBlock in a B-VOP (in MPEG-4 Advanced Simple Profile) where there are four Forward and four Backward motion vectors:



VOP no.	58
Pixel location	(224, 272)
MB location	(14, 17)
Address	0x01cdce (dec. 118222) bit 7
Mode	DIRECT
Quant	16
Bits	7
GMC	OFF
AC prediction	
CBP	0 (000000)
Forward MV	< -0.5, 0.0> < -0.5, 0.0> < -0.5, 0.0> < -0.5, 0.0>
Backward MV	< -0.5, 0.0> < -0.5, 0.0> < -0.5, 0.0> < -0.5, 0.0>

MB Statistics

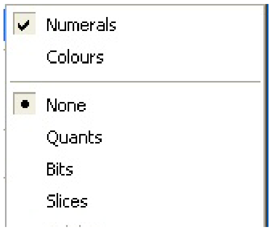
This option displays statistics on a MacroBlock-by-MacroBlock basis, either for the current frame only or averaged over a range of frames.

The top of this submenu allows selection of:

- Numerals (numbers)
- Colors

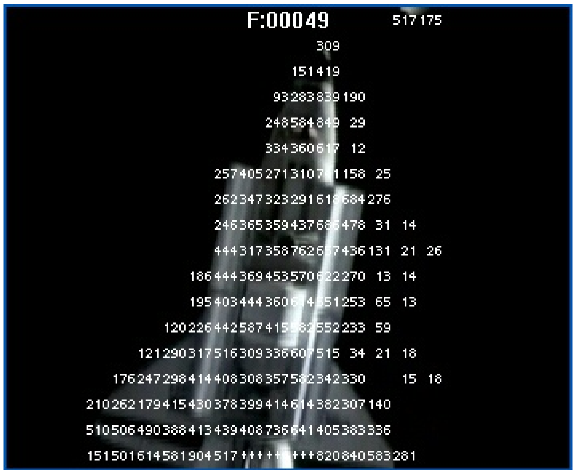
Numerals

When this is selected, numbers are displayed for the MacroBlock statistics, overlaid onto each MacroBlock.

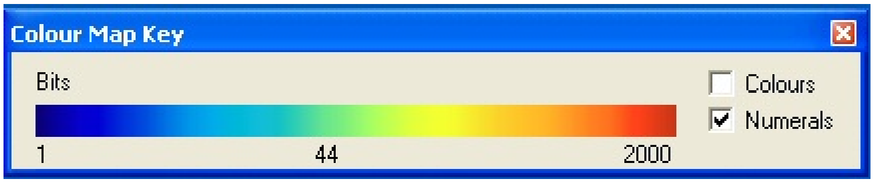


(In the preceding example, Numerals os selected, but None is also selected, so no statistics will be displayed.)

The displayed number is selected from this submenu; the following example shows bits per MacroBlock in frame 49 of the MPEG-4 example file Space (no bits = not coded):



When statistics have been selected to be displayed, the Color Map Key is also displayed:



☐ Colours
☑ Numerals

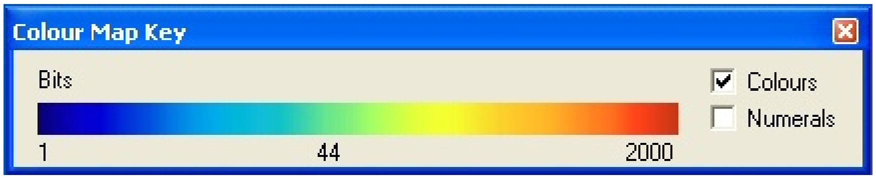
See the next section for a description of the Color Map Key.

NOTE. *In the Color Map Key, selecting the Colors displays the colors as well as the numerals.*

Colors

When this is selected, colors are displayed for the MacroBlock statistics, overlaid onto each MacroBlock.

The colors give a quick visual indication of the values in the video; the smaller the number, the more towards blue, the larger the number, the more towards red:



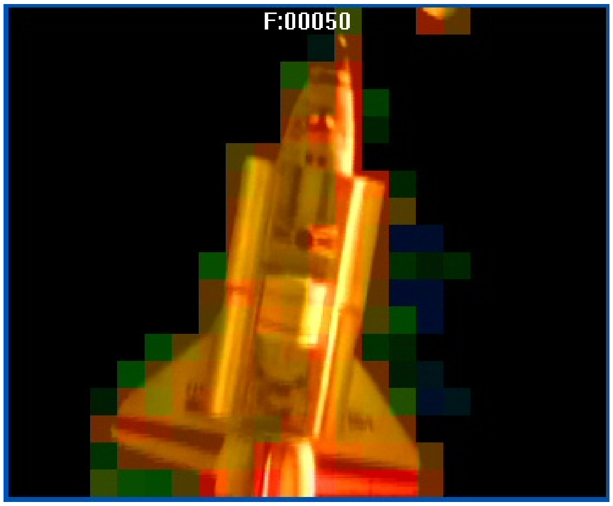
☑ Colours
☐ Numerals

NOTE. *If it is hard to see the colors, the video can be blanked (set to mid gray) by selecting Blank video in the Overlay menu. You can also blank the video by pressing the B key when the video is selected.*

If exact numbers are required, turn on the numerals, instead of , or in addition to, the colors, or use the MacroBlock tooltip.

If it is not displayed, the Color Map Key can also be displayed from Windows > Toolbars.

The example below shows bits per MacroBlock in frame 50 of the MPEG-4 example file Space (no color = no bits):



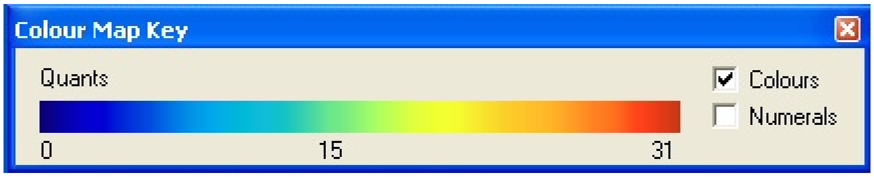
Linear/Logarithmic Scales

Some number ranges are best represented in linear scales (such as quants and slices); others are best in logarithmic scales.

MTS4EA automatically chooses an appropriate scale and displays the values at the bottom, middle, and top of the color range.

The choice of linear/logarithmic scale made by the MTS4EA is indicated by the middle value on the color key. If the value is not numerically the average of the top and bottom values, a logarithmic scale has been chosen by the MTS4EA.

Linear scale (for example, quants, slices):


Colour Map Key

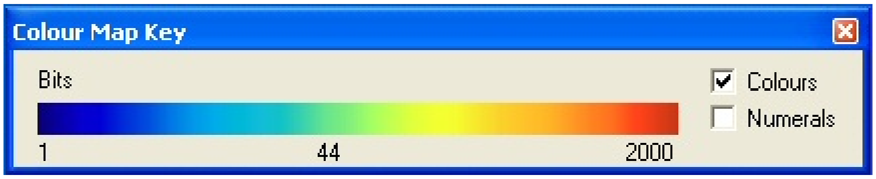
Quants

0 15 31

☑ Colours

☐ Numerals

Logarithmic scale (for example, bits, average bits):


Colour Map Key

Bits

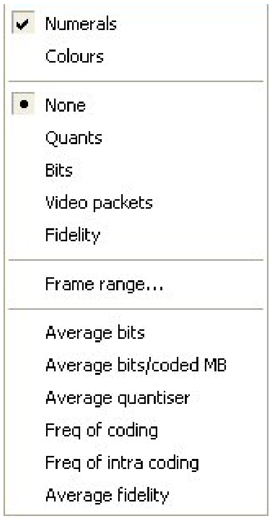
1 44 2000

☑ Colours

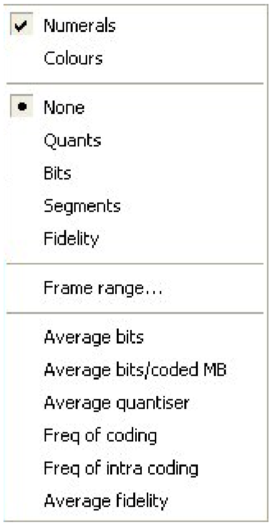
☐ Numerals

Menu Variation with Standard

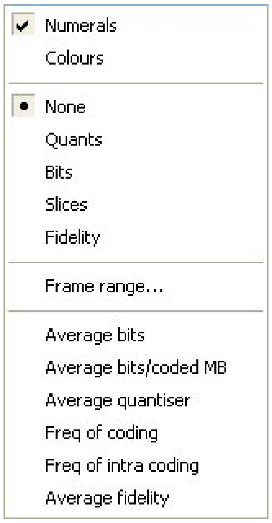
The menu below the selection of numerals/colors varies (slightly) depending on the video standard:



MPEG-4



H.263



H.264/AVC, MPEG-2

NOTE. *It can also useful to use this option with the Hold last frame option in the Play Menu, which allows the total sequence statistics to be viewed in relation to the picture content.*

For interlaced video streams, some of the MB statistics overlays are not visible if the video is displayed in combined view (frame view). To see the statistics in this situation, switch to the separated fields view using the interlace toolbar (see Separated fields view. *on page 6–276 for more information).*

The first four of the MB statistics overlays apply only to the current frame:

- Quants
- Bits
- Segments or Slices
- Fidelity

The remaining selections apply to the range of frames selected by Frame range.

- Average bits
- Average bits/coded MB
- Average quantizer
- Freq of coding
- Freq of intra coding
- Average fidelity

NOTE. ***T****he frame range over which statistics are to be accumulated cannot be changed during pause mode; otherwise MTS4EA could not ensure that the Trace and Graph data were collected over the correct range of frames. For example, if the video was paused at frame 23 and Trace was enabled for a range of frames from 20 to 25, the Trace data would be incorrect.*

When a range of frames has been set, the Frame range... item on the MB statistics menu changes, with a check mark to indicate that a range of frames has been set (see Frame Range on page 6–109).

The status bar indicates whether the video frame being displayed is in or out of the selected range of frames (see Frame range in/out indicator on status bar. on page 6–112).

None. This option switches off any displayed overlaid MacroBlock statistics on the current frame.

NOTE. *The None option does not turn off the motion vector plot or MacroBlock type display.*

Quants. This option shows the quantizer used for the luminance for each decoded MacroBlock. (The quantizer used for the Chrominance could be different, for example in H.263+ Annex T.)

If the MacroBlock was not coded, then no number/color is displayed.

This option can be used with Motion Vectors and/or MacroBlock types, but not with any other digit overlay option.

Bits. This option overlays the number of bits used to encode each MacroBlock in the current frame.

If the MacroBlock was not coded, no bits are shown/no color is shown, even though in MPEG-4 and H.263 a single bit is sent to signify an uncoded MacroBlock.

NOTE. *Because there is only enough space to display three digits, if the MacroBlock used more than 999 bits, then +++ is displayed to signify an overflow (a number greater than 999).*

This option can be used with Motion Vectors and/or MacroBlock types, but not with any other digit overlay option.

Video packets [MPEG-4]. This option shows the grouping of the MacroBlocks into video packets. The start of a new video packet is distinguished by a resynchronization marker in the stream.

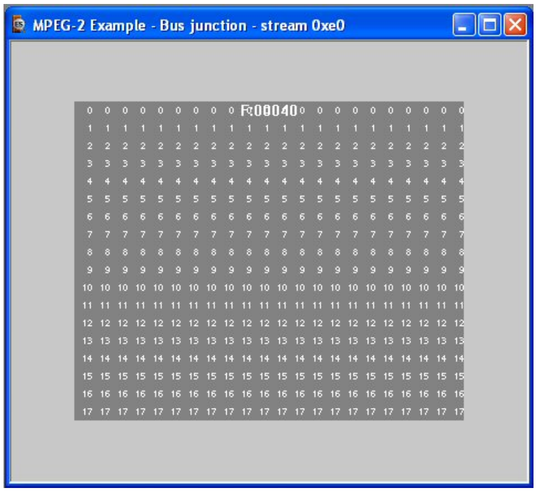
NOTE. *Video packets are only used in displays of information relating to MPEG-4.*

Segments [H.263]. This option shows the segment numbering in use for the current frame. Segments are defined as being Group of Blocks (if GOB headers are sent in H.263) or Slices (as defined in Annex K of H.263+). If a segment number is zero, then it is not displayed - hence if no GOB headers or Slices are sent, then this option will not overlay anything.

NOTE. *Segments are used only in displays of information relating to H.263.*

This option can be used with Motion Vectors and/or MacroBlock types, but not with any other digit overlay option.

Slices [H.264/AVC, MPEG-2]. This option shows the slice numbering in use for the current frame; an example is given below:

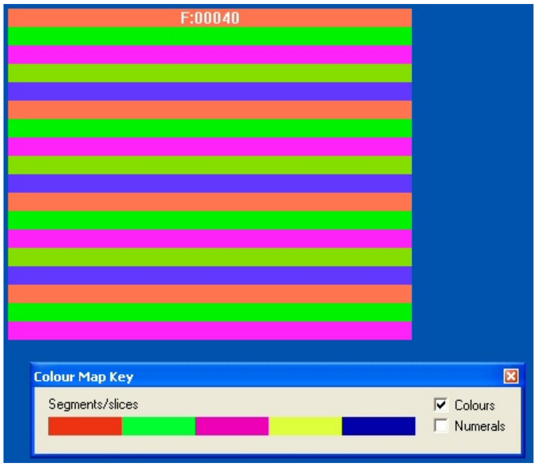


```
 0  0  0  0  0  0  0  0  0 F:00040 0  0  0  0  0  0  0  0  0  0
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
 3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
 4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4
 5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5  5
 6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6
 7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7  7
 8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8
 9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
```

NOTE. *Slices are used only in displays of information relating to H.264/AVC and MPEG-2.*

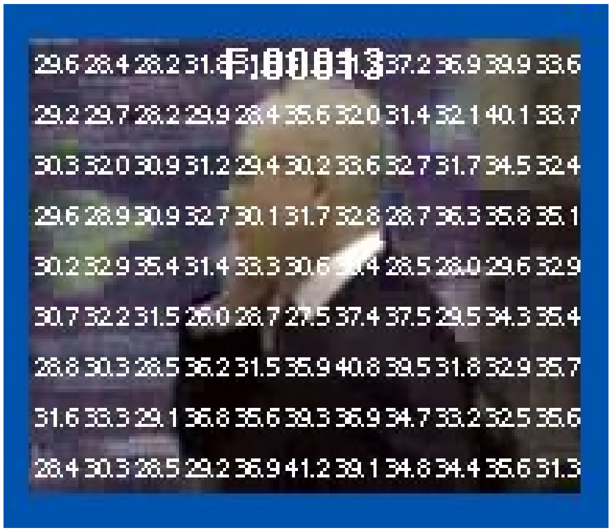
(In order to see the slice ID numbers more clearly, the video has been blanked, using the Overlay > Blank video menu selection.)

This shows the same frame, with the slices displayed using the color overlays:



The slice ID overlay option can be used with Motion Vectors and/or MacroBlock types, but not with any other digit overlay option.

Fidelity. This option shows the fidelity analysis values for each MacroBlock, for example, PSNR in dB (decibels):



29.6	28.4	28.2	[illegible]	[illegible]	[illegible]	[illegible]	37.2	36.9	39.9	33.6
29.2	29.7	28.2	29.9	28.4	35.6	32.0	31.4	32.1	40.1	33.7
30.3	32.0	30.9	31.2	29.4	30.2	33.6	32.7	31.7	34.5	32.4
29.6	28.9	30.9	32.7	30.1	31.7	32.8	28.7	36.3	35.8	35.1
30.2	32.9	35.4	31.4	33.3	30.6	[illegible]	28.5	28.0	29.6	32.9
30.7	32.2	31.5	26.0	28.7	27.5	37.4	37.5	29.5	34.3	35.4
28.8	30.3	28.5	36.2	31.5	35.9	40.8	39.5	31.8	32.9	35.7
31.6	33.3	29.1	36.8	35.6	39.3	36.9	34.7	33.2	32.5	35.6
28.4	30.3	28.5	29.2	36.9	41.2	39.1	34.8	34.4	35.6	31.3

NOTE. *Fidelity is grayed out until fidelity analysis is enabled; see* Video Fidelity Enable on page 6–162.

The type of fidelity analysis done is shown in the Fidelity tab of the Analysis options - see *Video Fidelity Enable* on page 6–162 for more information.

Frame Range

This option allows the start and end frames to be set for the calculation of the MacroBlock statistics (the range is inclusive):

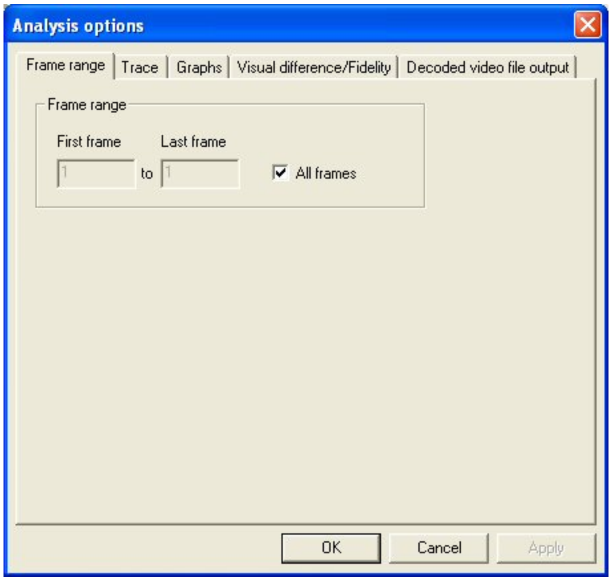
- Average bits
- Average bits/coded MB
- Average quantizer
- Freq of coding
- Freq of intra coding
- Average fidelity

NOTE. *The frame range set on this tab is the same frame range used for the Trace and Graph analysis (see Frame Range Tab on page 6–154).*

When the frame range has been set, the statistics are calculated from the first to the last frame within the range (inclusive frame numbers) and the last values they reach are held on the displayed frame when outside the range.

The frame range over which statistics are to be accumulated cannot be changed during pause mode; otherwise the MTS4EA could not ensure that the Trace and Graph data were collected over the correct range of frames. For example, if the video was paused at frame 23 and Trace was enabled for a range of frames from 20-25, the Trace data would be incorrect.

Clicking on Frame range opens the frame range tab:



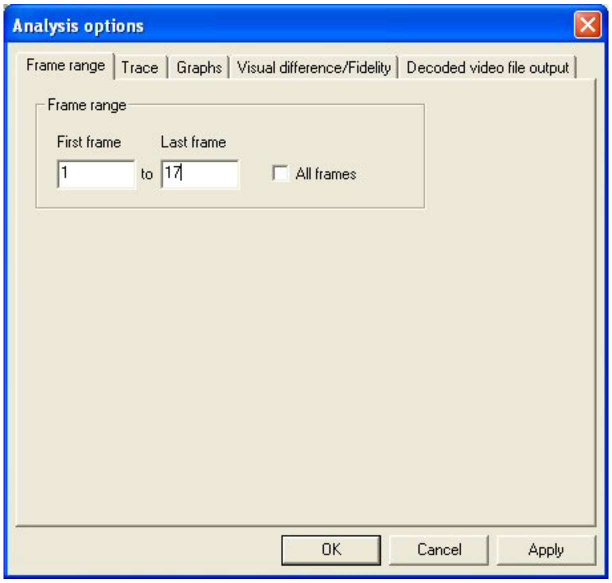
Frame range

First frame 1 to Last frame 1

☑ All frames

All frames. This option is the default and calculates the sequence statistics from the start of the sequence to the current frame.

To set a range of frames. If the All frames check box is cleared, then a range of frames can be entered (in this case, frames 1 to 17 inclusive):

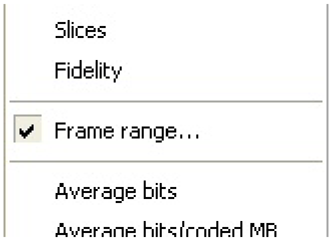


Frame range

First frame 1 to Last frame 17

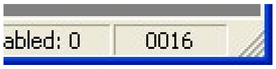
☐ All frames

When a range of frames has been set, the Frame range... item on the MB statistics menu changes, with a check mark to indicate that a range of frames has been set:

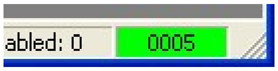


Frame range in/out indicator on status bar. The frame number indicator at the bottom right of the status bar changes when in/out of the frame range as follows:

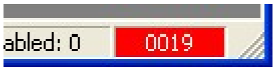
The frame counter text is in black with a gray background:



When a frame range has been set and the displayed frame is within the specified range, the frame counter background changes to green, as shown:



When a frame range has been set but the displayed frame is outside of the range set, the frame counter changes to white text on a red background.

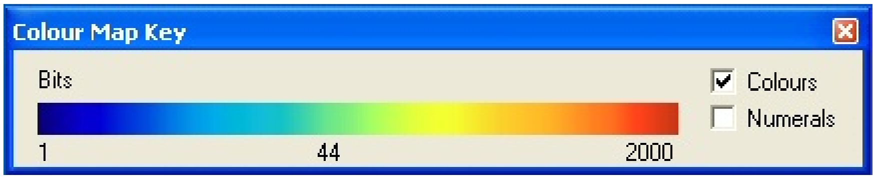


Average bits. This option displays the average number of bits used to code each MacroBlock, by dividing the total accumulated bits for each MacroBlock by the total number of frames in the frame range.

Both coded and not coded MacroBlocks are included in the average bits statistics.

NOTE. *Because there is only enough space to display three digits, if the MacroBlock used more than 999 bits, then +++ is displayed to signify an overflow (a number greater than 999).*

The color range displayed with the color overlays is a logarithmic scale:



☑ Colours

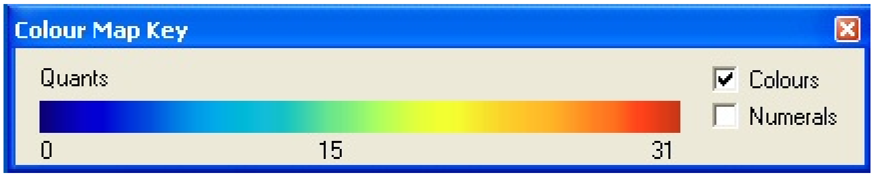
□ Numerals

Average bits/coded MB. This option displays the average number of bits used to code each MacroBlock, but allowing for the number of times it has been coded, the total accumulated bits used to code the MacroBlock over the frame range specified divided by the number of times that particular MacroBlock has been coded.

The color range displayed with the color overlays is a logarithmic scale: see preceding screenshot.

Average quantizer. This option displays the average quantizer used to code each MacroBlock over the specified frame range. It is displayed to one decimal place.

The color range displayed with the color overlays is a linear scale:



☑ Colours
☐ Numerals

Freq of coding. This option displays the frequency of coding (number of times each MacroBlock has been coded over the frame range specified).

NOTE. *Because there is only enough space to display three digits, if the MacroBlock used more than 999 bits, then +++ is displayed to signify an overflow (a number greater than 999).*

Freq of intra coding. This option displays the number of times each MacroBlock has been coded in Intra Mode over the frame range specified.

Average fidelity. This option shows the average fidelity analysis values for each MacroBlock over the selected frame range.

The type of fidelity analysis done is shown in the Fidelity tab of the Analysis options - see Video Fidelity Enable on page 6–162 for more information.

Overlay Color Ctrl+W

This item allows the selection of the color of the overlays. You can see the values even if the decoded image is very dark (choose white) or very light (choose black). It affects all the number overlays as well as the motion vector overlay.

In the motion vector overlay, two colors may be used for some video standards - in this case:

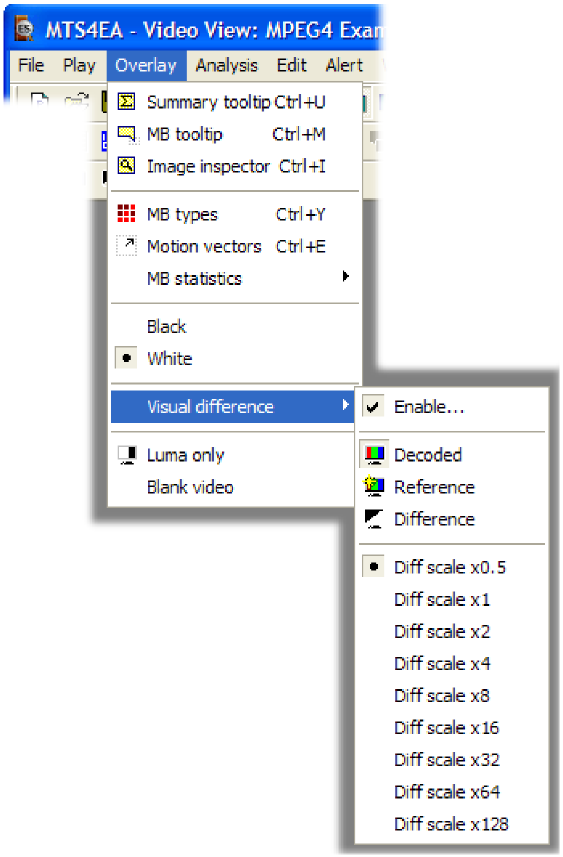
- Black selects black and dark blue colors for overlays
- White selects white and light blue colors for overlays

See *Motion Vectors* `Ctrl+E` on page 6–98 for more information.

Black. This selects a black/dark blue color for the overlays.

White. This selects a white/light blue color for the overlays.

Visual Difference



This menu option enables the visual difference video display; this uses an uncompressed video reference file to show a video view of any of the following:

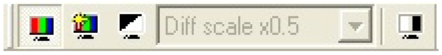
- The encoded (compressed) bitstream
- The uncompressed video reference bitstream
- The visual difference between the encoded bitstream and the uncompressed video reference file

NOTE. *The visual difference display can be done on a range of frames only, if this range is at the start of a file. For example, only the first ten frames of the YUV reference file are provided for the H.264/AVC and MPEG-2 bitstreams Grenadier Guards.*

When displaying the compressed bitstream or the uncompressed video reference file or the difference between the two, all the standard MTS4EA video playing controls can be used to play the video forward/backward (although some of the unrelated seek functions be grayed out).

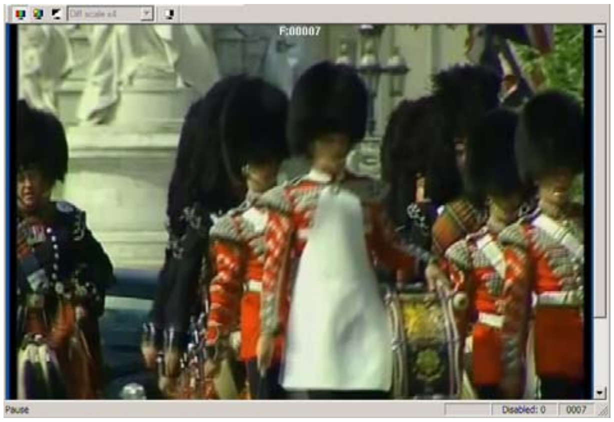
Visual Difference Icon Toolbar

This toolbar is displayed when the visual difference view is enabled:

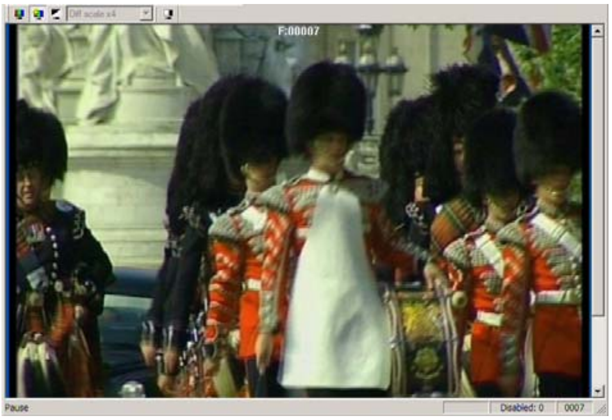


The functions of the icons are as follows:

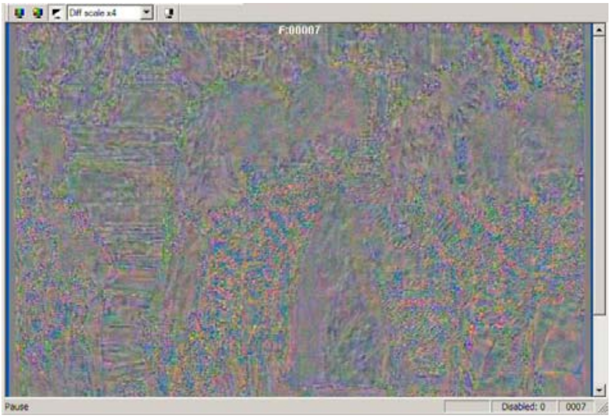
Show encoded (compressed). Selecting this menu option (or clicking on the 🖥 icon) displays the standard video window (shows the compressed bitstream that was decoded by the MTS4EA):



Show uncompressed video reference. Selecting this menu option (or clicking on the 🖥 icon) displays the frame in the YUV reference file that corresponds most closely in time to the corresponding frame in the compressed bitstream:



Show difference. Selecting this menu option (or clicking on the icon) subtracts the uncompressed video reference video from the compressed video, frame-by-frame:



The MTS4EA uses the timing given in the:

- Compressed bitstream itself for the displayed frame times of the compressed bitstream

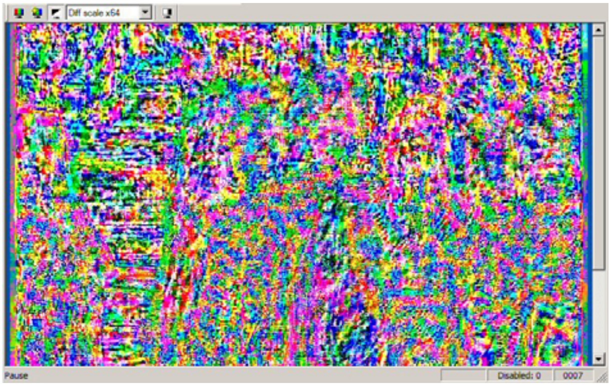
- Uncompressed video reference set-up tab of the MTS4EA (the frame rate) for the uncompressed video reference file

When doing the subtraction, the MTS4EA uses the uncompressed video reference file and the corresponding frames from the compressed bitstream that are closest to each other in time.

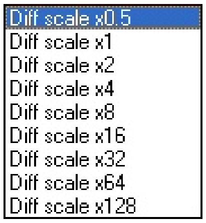
NOTE. *The visual difference view can be used to display the difference between two uncompressed video files; to do this, open the first uncompressed video file using the standard File-open (selecting Open as type) and then select the second uncompressed video as the reference file.*

If the frame rate of the uncompressed video reference file is set incorrectly, the visual differences will invariably appear to be much greater than they should be.

Magnify visual difference by X. This menu selection or drop-down menu can be used to magnify the visual differences on-screen, so that they are easier to see:



The magnification (multiplication factor) can be any of the factors shown:

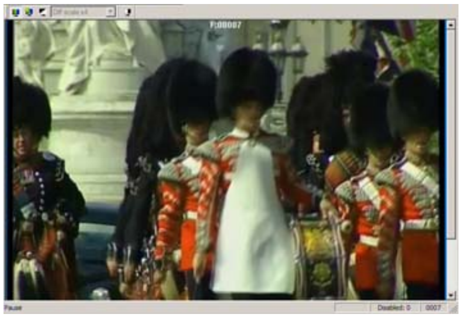


Diff scale x0.5
Diff scale x1
Diff scale x2
Diff scale x4
Diff scale x8
Diff scale x16
Diff scale x32
Diff scale x64
Diff scale x128

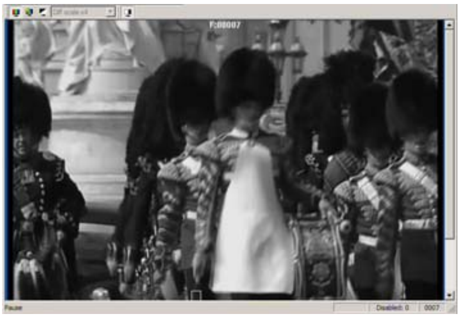
Luma Only

This displays the luma (luminance) only of the compressed bitstream, or uncompressed video reference, or difference.

When this option is selected, only the luminance of the image in the video window is shown:



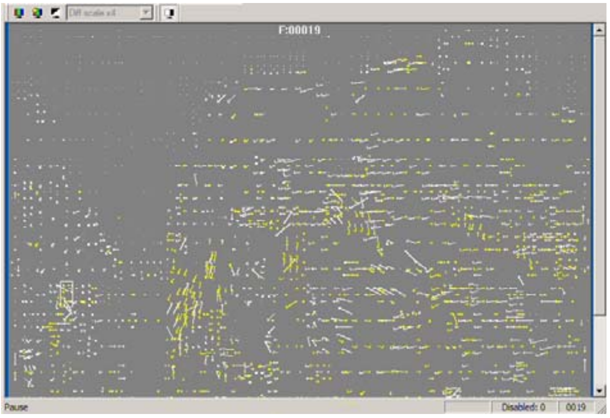
Luma only not selected



Luma only selected

Blank Video

This option will blank the video output to a gray color. It is included so that the motion vectors and other data can be viewed more clearly without the visual interference of the decoded image.



In this example, the motion vector overlay is on and the video is blanked. You can also blank the video by pressing the B key when the video is selected.

NOTE. *The Blank video option is available only when one of the overlays on the Overlay menu is in use; otherwise it is grayed out.*

Audio Options

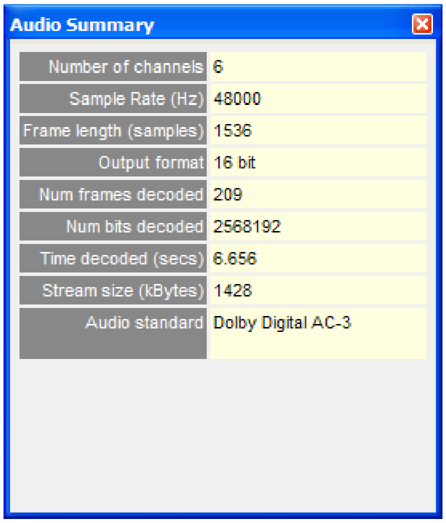
These options are only available when an audio waveform is being viewed.

NOTE. *Full audio analysis is not currently supported for the Dolby Digital (AC-3) standard. For this standard, only audio extraction and playback will be available.*

The audio summary and frame tooltips can be manipulated in the same way as the video tooltips (see Summary & Macroblock Tooltips Manipulation, page 6–71).

Audio Summary Tooltip

When this option is selected, a window is displayed that provides summary information about the audio frame being displayed or the sequence as a whole. The information displayed differs depending on whether the stream is playing/paused or is in stop mode. The selection of information provided by the audio summary tooltip is as follows:



Number of channels	6
Sample Rate (Hz)	48000
Frame length (samples)	1536
Output format	16 bit
Num frames decoded	209
Num bits decoded	2568192
Time decoded (secs)	6.656
Stream size (kBytes)	1428
Audio standard	Dolby Digital AC-3

Number of Channels

If the audio stream is mono, this number will be one (1); if the stream is stereo, this number will be two or more.

Sample Rate (Hz)

Sample rate of the decoded audio in samples per second (Hz).

Frame Length (samples)

Number of audio samples in each decoded audio frame.

Output Format

The output audio sample format bit depth in bits per sample.

Num Frames Decoded

The running total of decoded frames.

Total Frames

The total number of decoded audio frames in the stream.

Time Decoded (secs)

The time in seconds from the start to the current point in the sequence, calculated as if the sequence were playing normally (the MTS4EA keeps track of the playing time, irrespective of whether the sequence is paused at some point or played in fast forward mode).

Total Time (secs)

The duration of the stream in seconds.

Ave. Bit Rate (kbits/s)

This is total bits (kbits) divided by total time (secs).

Ave. Frame Rate (Hz)

This is total frames divided by total time (secs).

Stream Size (kBytes)

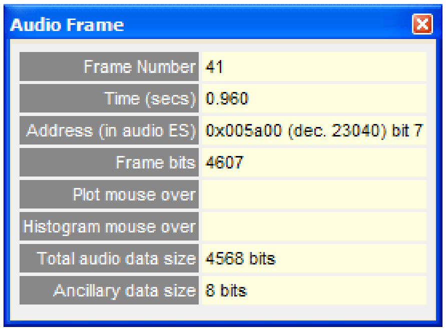
The size of the stream in KB.

Audio Standard

The audio standard used.

Audio Frame Tooltip

This tooltip displays information about a selected audio frame. The information provided by the frame tooltip varies according to the audio standard being analyzed and the options available for that standard.



Audio Frame

Frame Number	41
Time (secs)	0.960
Address (in audio ES)	0x005a00 (dec. 23040) bit 7
Frame bits	4607
Plot mouse over	
Histogram mouse over	
Total audio data size	4568 bits
Ancillary data size	8 bits

Frame Number

The number of the currently selected audio frame.

Time (secs)

The time in seconds from the start of the audio sequence to the start of the selected audio frame, calculated as if the sequence were playing normally (MTS4EA keeps track of the playing time, irrespective of whether the sequence is paused at some point or played in fast forward mode).

Address (in audio ES)

The address of the beginning of the selected audio frame within the audio elementary stream.

Frame Bits

The number of bits within the selected audio frame.

Plot Mouse Over

When the mouse pointer is over an audio plot, this field shows the name of the audio plot and its value for the selected audio frame.

Histogram Mouse Over

When the mouse pointer is over an audio histogram, this field shows the name of the part of the audio histogram the mouse is over and its value for the selected audio frame.

Total Audio Data Size [MPEG-1]

The total number of bits used for the audio data part of the selected frame. This consists of all the data that decodes to produce the audio frame, and does not include other data such as the header, error check, or ancillary data.

Ancillary Data Size [MPEG-1]

The total number of bits used for the ancillary data part of the selected audio frame.

Total SBR Data Size [AAC]

The total number of bits used for the SBR data part of the selected audio frame.

Total Config Data Size [AAC]

The total number of bits used for the configuration data part of the selected audio frame. This includes all data used to configure the AAC decoder (for example, ADTS headers and AudioSpecificConfig data.)

Total Data Size [AAC]

The total number of bits used to represent the main and subsidiary payload structures of the selected audio frame.

Max Scalefactor Blocks [AAC]

The maximum number of scalefactor bands per group for the selected audio frame.

Global Gain [AAC]

Global gain, equal to the value of the first scalefactor, for the selected audio frame.

LTP Data Present [AAC]

Flag to specify whether LTP data is present in the selected audio frame.

Predictor Data Present [AAC]

Flag to specify whether predictor data is present in the selected audio frame.

MS Mask Present [AAC]

Flag to specify whether MS mask is present in the selected audio frame.

Pulse Data Present [AAC]

Flag to specify whether pulse data is present in the selected audio frame.

TNS Data Present [AAC]

Flag to specify whether TNS data is present in the selected audio frame.

Gain Control Data Present [AAC]

Flag to specify whether gain control data is present in the selected audio frame.

Window Sequence [AAC]

Shows the type of window sequence used in the selected audio frame.

Window Shape [AAC]

Shows the type of window shape used in the selected audio frame.

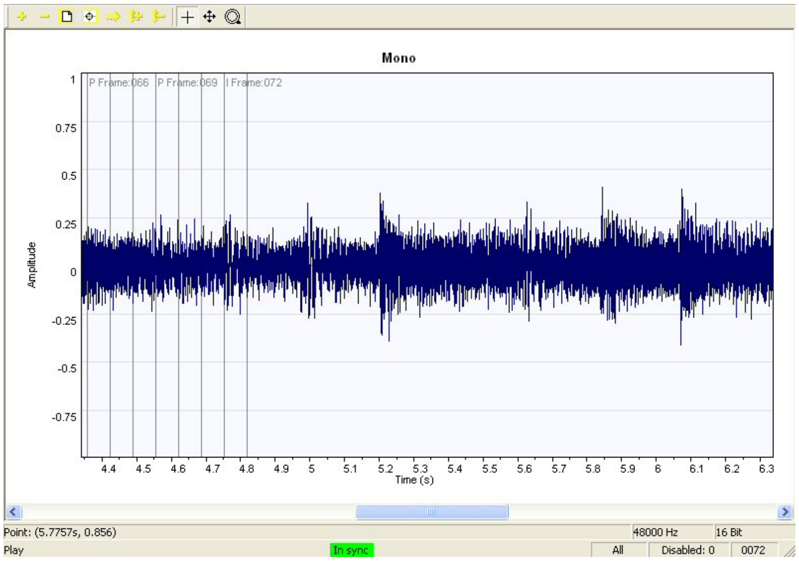
Show Audio Features...

A range of options showing audio related data.

Show Audio waveform ()

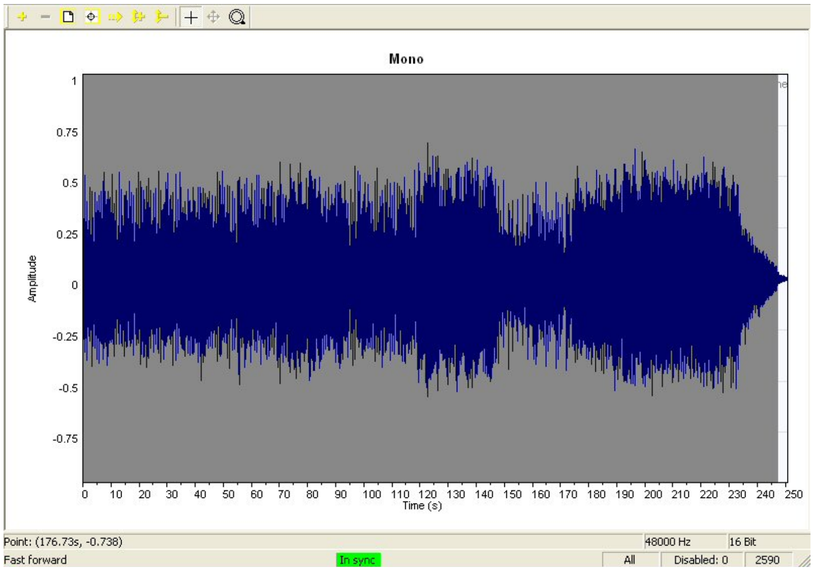
MTS4EA offers the ability to view an audio streams waveform: the y-axis is normalized amplitude, and the x-axis is time (seconds).

The audio streams waveform can be viewed while the stream is playing, showing the frames being marked off on vertical bars as the stream is decoded:



Point: (5.7757s, 0.856) 48000 Hz 16 Bit
Play All Disabled: 0 0072

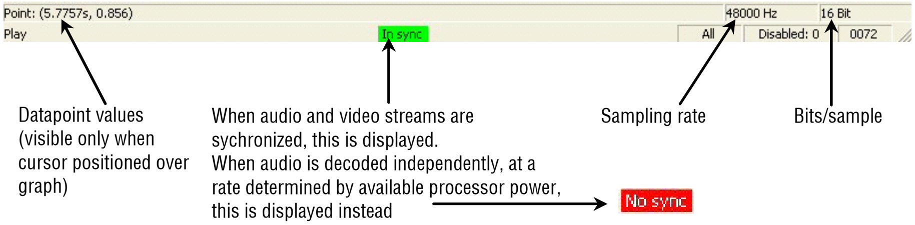
It can also be viewed when the stream has been played through to its end:



Point: (176.73s, -0.738) 48000 Hz 16 Bit
Fast forward All Disabled: 0 2590

The gray background is due to the density of marked-off frames; when the zoom out icon is used, these will become more spaced and the shape of the waveform more easily discernible (see the following figure). At each frame marker, the frame type and number is shown.

Additional information can be found in the Audio waveform views status bar:



Point: (5.7757s, 0.856) | 48000 Hz | 16 Bit

Play | In sync | All | Disabled: 0 | 0072

Datapoint values (visible only when cursor positioned over graph)

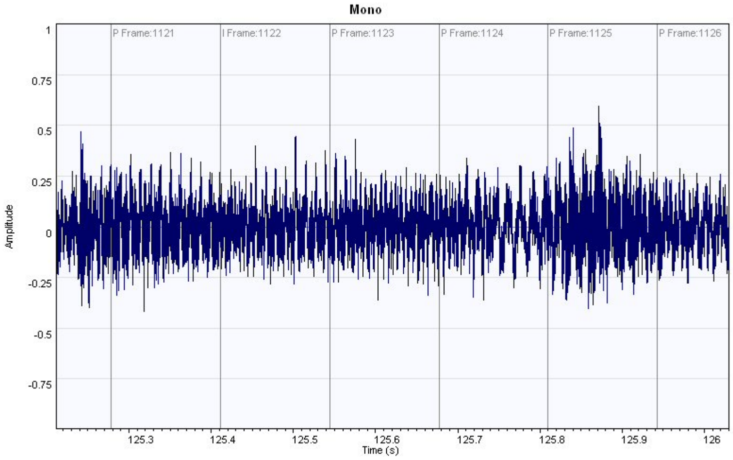
When audio and video streams are sychronized, this is displayed.
When audio is decoded independently, at a rate determined by available processor power, this is displayed instead

No sync

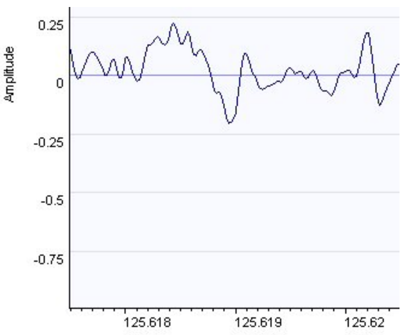
Sampling rate

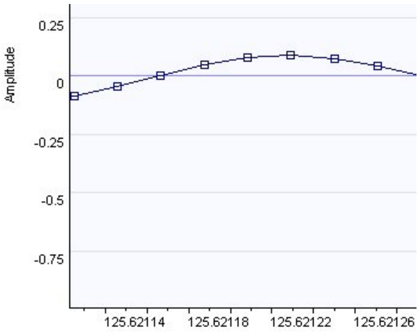
Bits/sample

It is possible to zoom in and out on this graph view using the first two icons on the waveform view toolbar, as illustrated below.



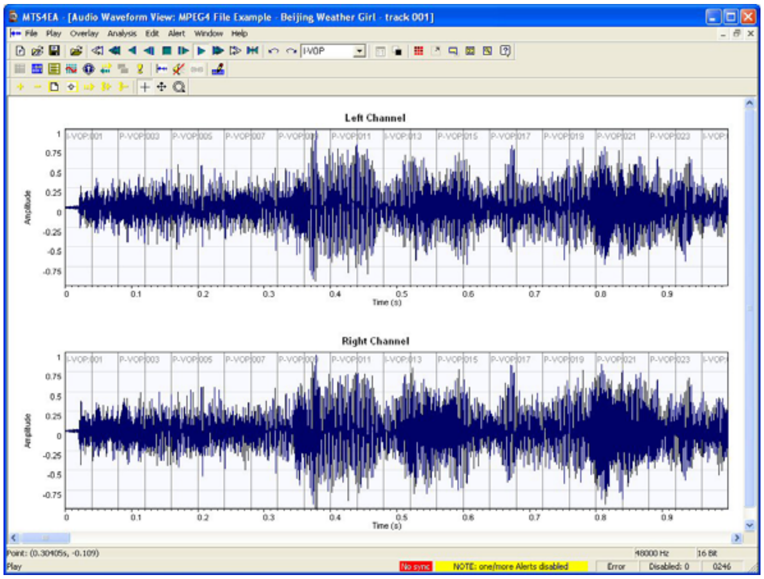
The illustrations below show that it is possible to zoom in so far as to see the individual data points plotted in the audio waveform.





The functions of the other icons are described under **Audio** Waveform Toolbar icons on page 6–127.

The following screenshot shows the waveform of a stream with multiple channels:


MTS4EA - [Audio Waveform View: MPEG4 File Example - Beijing Weather Girl - track 001]

Point: (0.30405s, -0.109)

48000 Hz

16 Bit

Play

No sync

NOTE: one/more Alerts disabled

Error

Disabled: 0

0246

Audio Waveform Toolbar icons

Icon	Function
+ −	Zoom in (+) and zoom out (-) centered on the window (affects scale of x-axis only).
▢	Fit all data into window.
⊕	Locate origin (zero), for example, start of sequence.
➡	Autoscroll, for example, fill the analysis data in real-time as the audio is decoded, and scroll the window to the right.
↕+	Increase channel height (affects scale of y-axis only).
↕−	Decrease channel height (affects scale of y-axis only).
+	Measure the data at the cursor. Values are reported in status bar. Hold and drag to measure offsets and angles/slopes of lines.
✥	Scroll/pan (the cursor changes to show the scroll/pan direction).
🔍	Zoom in/out centered on the cursor. Press <Shift> to zoom out.
|||	Show or hide audio frame markers.
📈	Show or hide the plots and plot key. See page 6–133 for a full account of analysis with the plot tool.
📊	Show or hide the selected histogram and associated key. See page 6–128 for a full account of histographical analysis options.
|↔	Show or hide the audio waveform.
|¹|²	Show or hide the audio/video synchronisation markers.

Show audio frame markers (|||)

Shows/hides the audio frame markers.

Show sync markers (|¹|²)

Shows/hides the audio synchronization markers.

Show Histogram

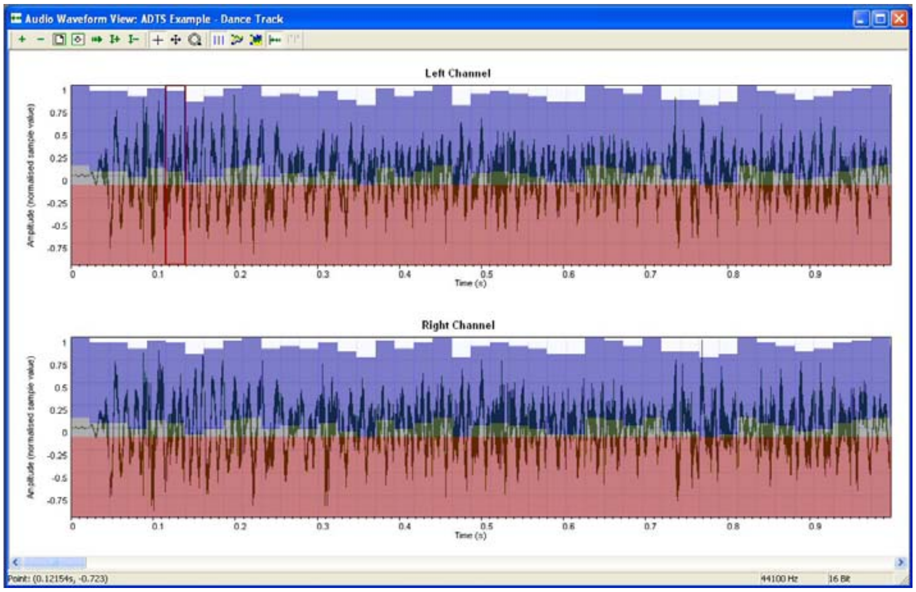
Shows/hides the distribution of different types of coded data within each audio frame. The available histograms depend on the audio standard in use. Only one type of histogram can be overlaid onto the audio waveform at any one time. The following histograms are available:

Audio Data Bit Usage Histogram [MPEG-1/2]

Shows the distribution of bits used for the following audio data syntax elements within each frame:

- allocation
- sample
- samplecode
- scalefactor
- scfsi

See *Compression Standards and File Types* section for a full definition of these syntax elements.


Audio Waveform View: ADTS Example - Dance Track

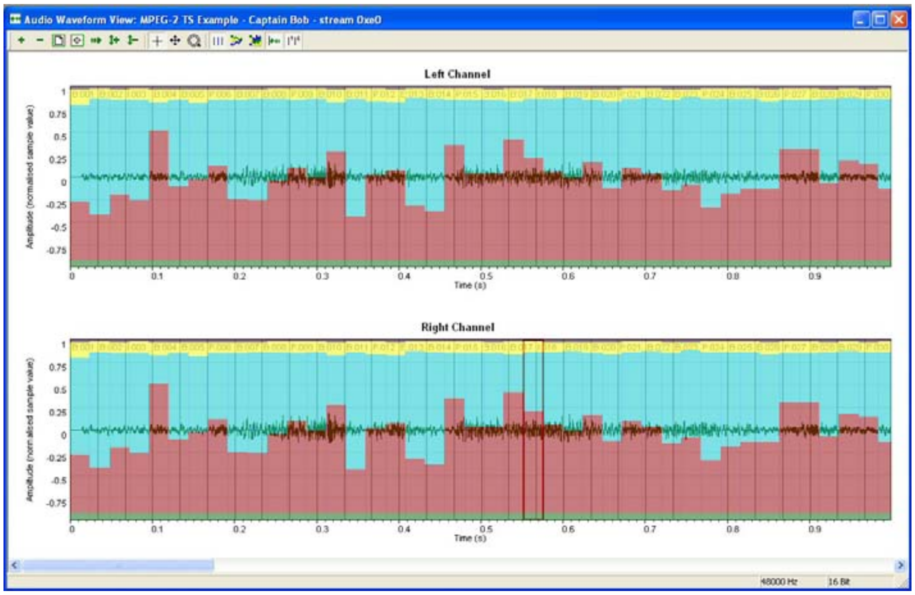
Point: (0.12154s, -0.723) 44100 Hz 16 Bit

Frame Data Bit Usage Histogram [MPEG-1/2]

Shows the distribution of bits used for the following syntactic structures within each frame:

- AncillaryData
- AudioData
- AudioSequence
- ErrorCheck
- Frame
- Header
- HuffmanCodeBits
- MainData

See *Compression Standards and File Types* section for a full definition of these syntax elements.


Audio Waveform View: MPEG-2 TS Example - Captain Bob - stream 0xe0
Left Channel
Amplitude (normalised sample value)
1
0.75
0.5
0.25
0
-0.25
-0.5
-0.75
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Time (s)
Right Channel
Amplitude (normalised sample value)
Time (s)
48000 Hz
16 Bit

Frame Bit Usage Histogram [AAC]

Shows the distribution of bits used for the following top-level categories of data within each frame:

- Config
- Main payload
- SBR (Spectral Band Replication)
- Subsidiary payload

See *Compression Standards and File Types* section for a full definition of these syntax elements.

Main Payload Bit Usage Histogram [AAC]

Shows the distribution of bits used for the following syntax elements within the main payload data of each frame:

- channel_pair_element
- coupling_channel_element
- data_stream_element
- fill_element
- gain_control_data
- ics_info
- lfe_channel_element
- program_config_element
- pulse_data
- raw_data_block
- single_channel_element

See *Compression Standards and File Types* section for a full definition of these syntax elements.

SBR Bit Usage Histogram [AAC]

Shows the distribution of bits used for the following syntax elements within the SBR data of each frame:

- sbr_channel_pair_base_element
- sbr_channel_pair_element
- sbr_channel_pair_enhance_element
- sbr_data
- sbr_dtdf
- sbr_envelope
- sbr_extension
- sbr_grid
- sbr_header
- sbr_invf
- sbr_noise
- sbr_single_channel_element
- sbr_sinusoidal_coding
- Unused SBR data when no header has been found

See *Compression Standards and File Types* section for a full definition of these syntax elements.

Subsidiary Payload Bit Usage Histogram [AAC]

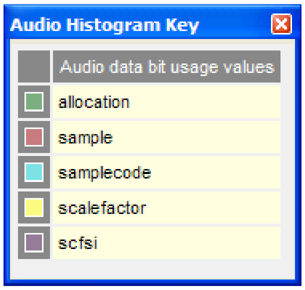
Shows the distribution of bits used for the following syntax elements within the subsidiary payload data of each frame:

- dynamic_range_info
- extension_payload
- individual_channel_stream
- ltp_data
- reordered_spectral_data
- scale_factor_data
- section_data
- spectral_data
- tns_data

See *Compression Standards and File Types* section for a full definition of these syntax elements.

Show Histogram Key

Shows/hides the currently selected histogram and the associated key; the first available histogram is displayed if there is no current selection. The fields listed in the key will vary according to which histogram is displayed.



Audio Histogram Key

	Audio data bit usage values
🟩	allocation
🟥	sample
🟦	samplecode
🟨	scalefactor
🟪	scfsi

If a key is open when the histogram overlay is changed, the key will be automatically updated.

Show plot

This menu presents you with a choice of different line chart overlays. The available line plots will depend on the specific audio standard in use. Any combination of the available plots can be plotted at any one time. The plots available for each standard are as follows:

Total frame size [All standards]: The total number of bits used for all data of each audio frame.

Total audio data size [MPEG-1/2]: The total number of bits used for the audio data part of each frame. This consists of all the data that decodes to produce the audio frame, and does not include other data such as the header, error check and ancillary data.

Ancillary data size [MPEG-1/2]: The total number of bits used for the ancillary data part of each audio frame.

Total SBR data size [AAC]: The total number of bits used for the SBR data part of each audio frame.

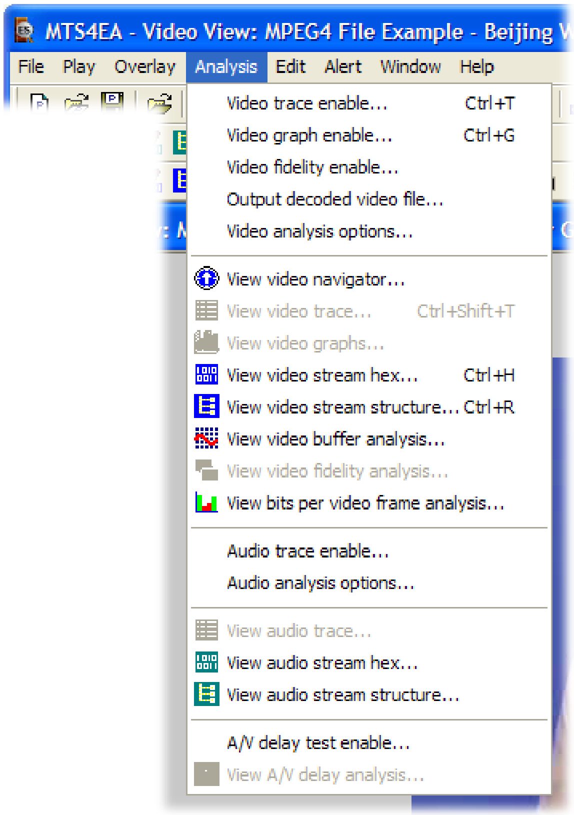
Total config data size [AAC]: The total number of bits used for the configuration data part of each audio frame. This includes all data used to configure the AAC decoder (for example, ADTS headers, AudioSpecificConfig data).

Total data size [AAC]: The total number of bits used to represent the main and subsidiary payload structures of each audio frame.

Max scalefactor blocks [AAC]: The maximum number of scalefactor bands per group for each audio frame.

Global gain [AAC]: Global gain, equal to the value of the first scalefactor, for each audio frame.

Analysis Menu


MTS4EA - Video View: MPEG4 File Example - Beijing V
File
Play
Overlay
Analysis
Edit
Alert
Window
Help
Video trace enable... Ctrl+T
Video graph enable... Ctrl+G
Video fidelity enable...
Output decoded video file...
Video analysis options...
View video navigator...
View video trace... Ctrl+Shift+T
View video graphs...
View video stream hex... Ctrl+H
View video stream structure... Ctrl+R
View video buffer analysis...
View video fidelity analysis...
View bits per video frame analysis...
Audio trace enable...
Audio analysis options...
View audio trace...
View audio stream hex...
View audio stream structure...
A/V delay test enable...
View A/V delay analysis...

This menu controls the collection and display of:

- Trace information, such as Trace/Parse bitstream and Trace/Interpret
- Graph information
- Fidelity analysis

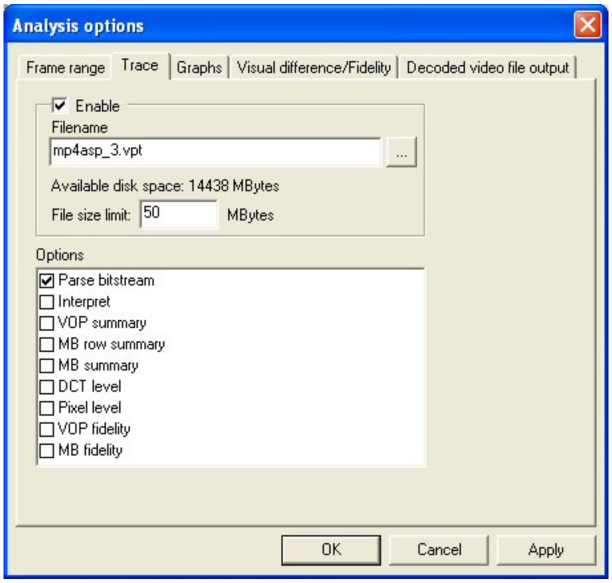
and the display of:

- Hex (hexadecimal) view of stream data
- The structure of the file being analyzed
- Analysis of the video buffer usage

The Analysis options menu item provides access to all the Trace, Graph and Fidelity options as well as setting of the range of frames over which to collect the Trace, Graph and Fidelity information.

NOTE. *Trace enable..., Graph enable..., Fidelity enable..., Decoded file output enable... and Analysis options... cannot be set in pause mode - the video must be stopped, otherwise MTS4EA could not ensure that the Trace, Graph and Fidelity data were collected over the correct range of frames.*

Video Trace Enable Ctrl+T



☑ Enable

Filename: mp4asp_3.vpt

Available disk space: 14438 MBytes

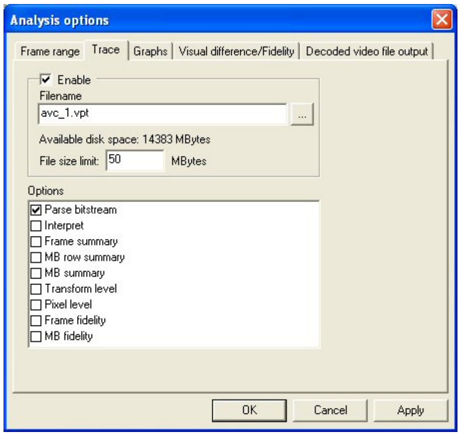
File size limit: 50 MBytes

Options

☑ Parse bitstream
☐ Interpret
☐ VOP summary
☐ MB row summary
☐ MB summary
☐ DCT level
☐ Pixel level
☐ VOP fidelity
☐ MB fidelity

The Trace options provide text outputs that describe the contents of an encoded video stream, to various levels of detail.

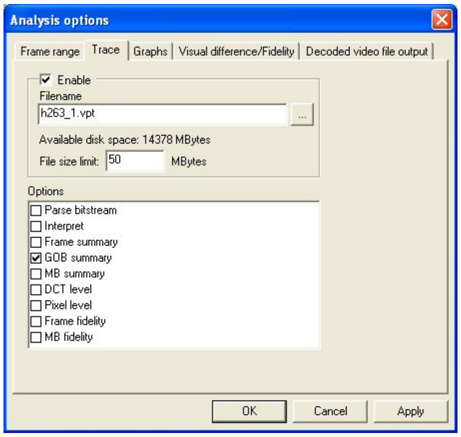
The available Trace options vary with each video standard; the screenshot above is for MPEG-4; the screenshots below for H.264/AVC and H.263 respectively.



☑ Enable
Filename: avc_1.vpt
Available disk space: 14383 MBytes
File size limit: 50 MBytes

Options
☑ Parse bitstream
□ Interpret
□ Frame summary
□ MB row summary
□ MB summary
□ Transform level
□ Pixel level
□ Frame fidelity
□ MB fidelity

H.264/AVC



☑ Enable
Filename: h263_1.vpt
Available disk space: 14378 MBytes
File size limit: 50 MBytes

Options
□ Parse bitstream
□ Interpret
□ Frame summary
☑ GOB summary
□ MB summary
□ DCT level
□ Pixel level
□ Frame fidelity
□ MB fidelity

H.263

NOTE. *Using these options can generate a lot of data - files hundreds of MB or more in size, so it is often advisable to collect the data on a range of frames, not all (this is a selectable option).*

Alternatively, you can set a maximum file size, after which no more trace data is output to the file concerned.
See *File size limit (available disk space)* on page 6–140.

NOTE. *The video can be stopped and the Trace file opened immediately after going past the last frame number in the frame range – there is no need to wait until the end of the video sequence.*

To set the range of frames over which to collect the Trace data, click on the Frame range tab.

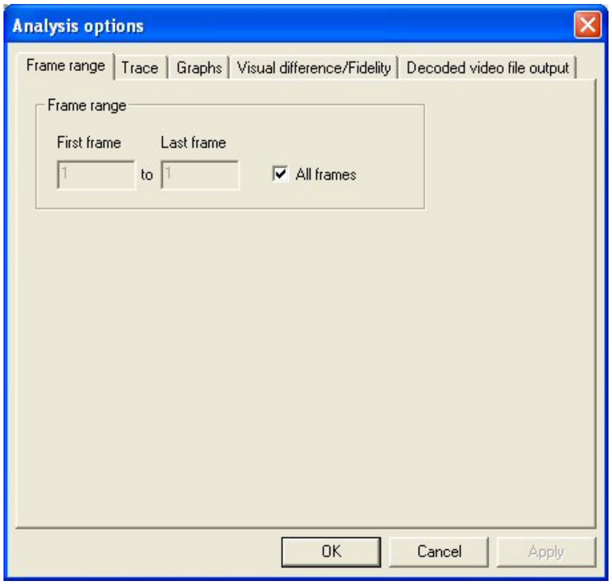
Frame Range Tab

This option allows you to specify the start and end frames between which to gather the Trace and Graphs information.

NOTE. *The frame range set on this tab is the same frame range used for the collection of MacroBlock statistics, where these are collected over a range of frames (see* Frame range *on page 6–109).*

The frame range over which statistics are to be accumulated cannot be changed during pause mode; otherwise the MTS4EA could not ensure that the Trace and Graph data were collected over the correct range of frames. For example, if the video was paused at frame 23 and Trace was enabled for a range of frames from 20-25, the Trace data would be incorrect.

Clicking on Frame range takes you to the frame range tab:

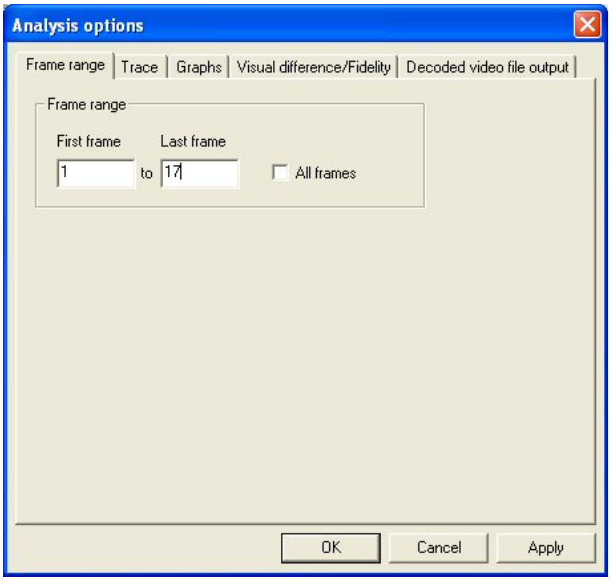


Frame range

First frame 1 to Last frame 1 ☑ All frames

All frames. This option is the default and calculates the sequence statistics from the start of the sequence until the current frame.

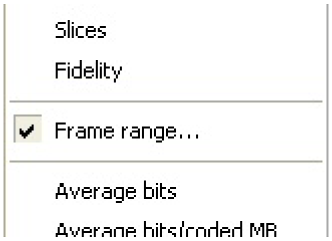
To set a range of frames. If the All frames check box is cleared, then a range of frames can be entered (in this case, frames 1 to 17 inclusive):



Frame range

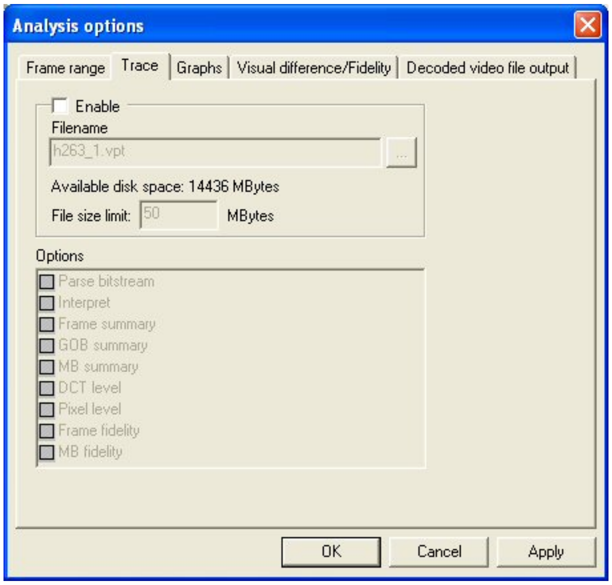
First frame 1 to Last frame 17 □ All frames

When a range of frames has been set, the Frame range... item on the MB statistics menu changes, with a check mark to indicate a range of frames has been set:



Trace Tab

Enable. This option Enables/Disables the Trace output. If the output is disabled, the remaining selections on the Trace tab are unavailable (grayed out) although the values are stored during the current use of the MTS4EA. (They are reset to their default values the next time the MTS4EA is started.)



□ Enable

Filename: h263_1.vpt

Available disk space: 14436 MBytes

File size limit: 50 MBytes

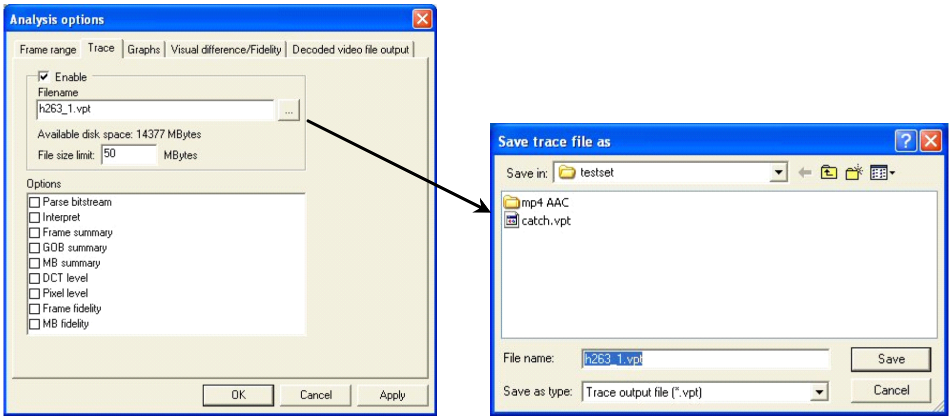
Options

□ Parse bitstream
□ Interpret
□ Frame summary
□ GOB summary
□ MB summary
□ DCT level
□ Pixel level
□ Frame fidelity
□ MB fidelity

Filename. By default, the suggested filename is the same as the base name of the input video file (the video filename without the extension).

Any other filename can be entered; the MTS4EA will append a `.vpt` extension to this filename if it does not already end in this. (.vpt = MTS4EA trace).

To browse to a specific folder and enter the filename there, click on the [...] button:



☑ Enable

Filename: h263_1.vpt ...

Available disk space: 14377 MBytes

File size limit: 50 MBytes

Options

☐ Parse bitstream
☐ Interpret
☐ Frame summary
☐ GOB summary
☐ MB summary
☐ DCT level
☐ Pixel level
☐ Frame fidelity
☐ MB fidelity

Save in: testset

mp4 AAC
catch.vpt

File name: h263_1.vpt

Save as type: Trace output file (*.vpt)

File size limit (available disk space). Trace files can generate a lot of data: for example, 2-3 MB per frame for the Parse bitstream and Interpret outputs. This means that collecting data over a large number of frames can easily generate a Trace file hundreds of MB in size.

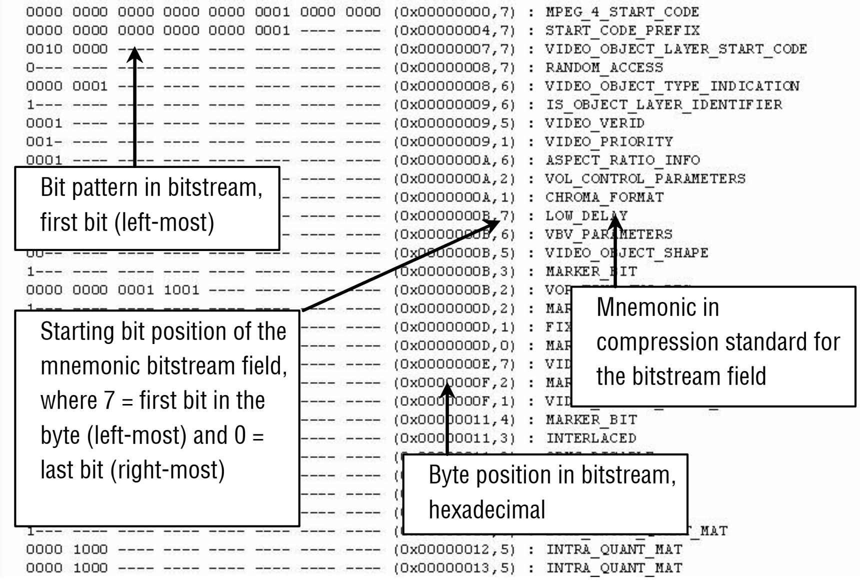
This entry allows you to limit the amount of disk space taken by the Trace file.

Once the Trace file reaches this size, it stops writing more data.

Parse Bitstream. If this option is selected, then the bitstream is parsed to a file. This gives the individual bit patterns for the bitstream fields as well as the mnemonic used within the standard to identify the field.

NOTE. *The data from the output of Parse Bitstream is one of the best ways that the MTS4EA provides of doing bitstream syntax debugging.*

See General codes used in Trace files and Alerts in the Bitstream Syntax Debugging *section for detailed information on the meaning of these.*



```
0000 0000 0000 0000 0000 0001 0000 0000 (0x00000000,7) : MPEG_4_START_CODE
0000 0000 0000 0000 0000 0001 ---- ---- (0x00000004,7) : START_CODE_PREFIX
0010 0000 ---- ---- ---- ---- ---- ---- (0x00000007,7) : VIDEO_OBJECT_LAYER_START_CODE
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000008,7) : RANDOM_ACCESS
0000 0001 ---- ---- ---- ---- ---- ---- (0x00000008,6) : VIDEO_OBJECT_TYPE_INDICATION
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000009,6) : IS_OBJECT_LAYER_IDENTIFIER
0001 ---- ---- ---- ---- ---- ---- ---- (0x00000009,5) : VIDEO_VERID
001- ---- ---- ---- ---- ---- ---- ---- (0x00000009,1) : VIDEO_PRIORITY
0001 ---- ---- ---- ---- ---- ---- ---- (0x0000000A,6) : ASPECT_RATIO_INFO
                              - ---- ---- (0x0000000A,2) : VOL_CONTROL_PARAMETERS
                              - ---- ---- (0x0000000A,1) : CHROMA_FORMAT
                              - ---- ---- (0x0000000B,7) : LOW_DELAY
                              - ---- ---- (0x0000000B,6) : VBV_PARAMETERS
00-- ---- ---- ---- ---- ---- ---- ---- (0x0000000B,5) : VIDEO_OBJECT_SHAPE
1--- ---- ---- ---- ---- ---- ---- ---- (0x0000000B,3) : MARKER_BIT
0000 0000 0001 1001 ---- ---- ---- ---- (0x0000000B,2) : VOP
1--- ---- ---- ---- ---- ---- ---- ---- (0x0000000D,2) : MAR
                            ---- ---- (0x0000000D,1) : FIX
                            ---- ---- (0x0000000D,0) : MAR
                            ---- ---- (0x0000000E,7) : VID
                            ---- ---- (0x0000000F,2) : MAR
                            ---- ---- (0x0000000F,1) : VID
                            ---- ---- (0x00000011,4) : MARKER_BIT
                            ---- ---- (0x00000011,3) : INTERLACED
                            ---- ---- (
                            ---- ---- (
                            ---- ---- (
                            ---- ---- (
1--- ---- ---- ---- ---- ---- ---- ---- (                            _MAT
0000 1000 ---- ---- ---- ---- ---- ---- (0x00000012,5) : INTRA_QUANT_MAT
0000 1000 ---- ---- ---- ---- ---- ---- (0x00000013,5) : INTRA_QUANT_MAT
```

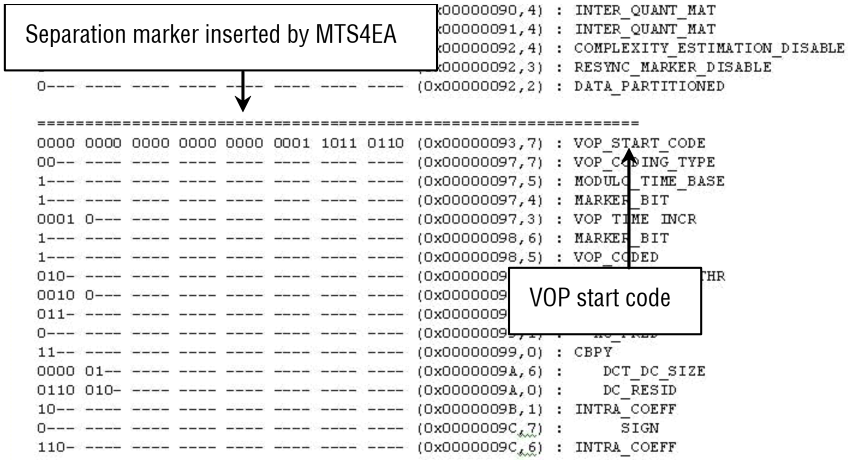
Bit pattern in bitstream, first bit (left-most)

Starting bit position of the mnemonic bitstream field, where 7 = first bit in the byte (left-most) and 0 = last bit (right-most)

Mnemonic in compression standard for the bitstream field

Byte position in bitstream, hexadecimal

MPEG-4 example, at start of bitstream

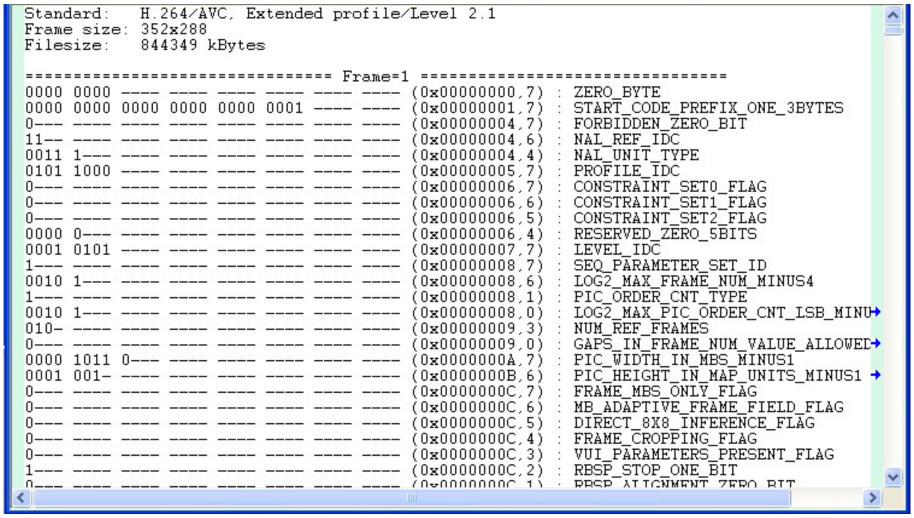


```
                                         x00000090,4) : INTER_QUANT_MAT
                                         x00000091,4) : INTER_QUANT_MAT
                                         x00000092,4) : COMPLEXITY_ESTIMATION_DISABLE
                                         x00000092,3) : RESYNC_MARKER_DISABLE
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000092,2) : DATA_PARTITIONED

=============================================================
0000 0000 0000 0000 0000 0001 1011 0110 (0x00000093,7) : VOP_START_CODE
00-- ---- ---- ---- ---- ---- ---- ---- (0x00000097,7) : VOP_CODING_TYPE
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000097,5) : MODULO_TIME_BASE
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000097,4) : MARKER_BIT
0001 0--- ---- ---- ---- ---- ---- ---- (0x00000097,3) : VOP TIME INCR
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000098,6) : MARKER_BIT
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000098,5) : VOP_CODED
010- ---- ---- ---- ---- ---- ---- ---- (0x0000009                  THR
0010 0--- ---- ---- ---- ---- ---- ---- (0x0000009
011- ---- ---- ---- ---- ---- ---- ---- (0x0000009
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000009
11-- ---- ---- ---- ---- ---- ---- ---- (0x00000099,0) : CBPY
0000 01-- ---- ---- ---- ---- ---- ---- (0x0000009A,6) :    DCT_DC_SIZE
0110 010- ---- ---- ---- ---- ---- ---- (0x0000009A,0) :    DC_RESID
10-- ---- ---- ---- ---- ---- ---- ---- (0x0000009B,1) : INTRA_COEFF
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000009C,7) :       SIGN
110- ---- ---- ---- ---- ---- ---- ---- (0x0000009C,6) : INTRA_COEFF
```

Separation marker inserted by MTS4EA

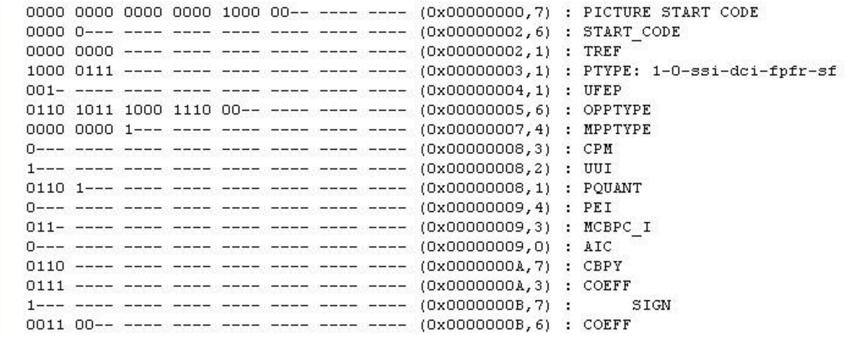
VOP start code

MPEG-4 example, at start of VOP



```
Standard:   H.264/AVC, Extended profile/Level 2.1
Frame size: 352x288
Filesize:   844349 kBytes

================================ Frame=1 ================================
0000 0000 ---- ---- ---- ---- ---- ---- (0x00000000,7) : ZERO_BYTE
0000 0000 0000 0000 0000 0001 ---- ---- (0x00000001,7) : START_CODE_PREFIX_ONE_3BYTES
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000004,7) : FORBIDDEN_ZERO_BIT
11-- ---- ---- ---- ---- ---- ---- ---- (0x00000004,6) : NAL_REF_IDC
0011 1--- ---- ---- ---- ---- ---- ---- (0x00000004,4) : NAL_UNIT_TYPE
0101 1000 ---- ---- ---- ---- ---- ---- (0x00000005,7) : PROFILE_IDC
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000006,7) : CONSTRAINT_SET0_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000006,6) : CONSTRAINT_SET1_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000006,5) : CONSTRAINT_SET2_FLAG
0000 0--- ---- ---- ---- ---- ---- ---- (0x00000006,4) : RESERVED_ZERO_5BITS
0001 0101 ---- ---- ---- ---- ---- ---- (0x00000007,7) : LEVEL_IDC
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000008,7) : SEQ_PARAMETER_SET_ID
0010 1--- ---- ---- ---- ---- ---- ---- (0x00000008,6) : LOG2_MAX_FRAME_NUM_MINUS4
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000008,1) : PIC_ORDER_CNT_TYPE
0010 1--- ---- ---- ---- ---- ---- ---- (0x00000008,0) : LOG2_MAX_PIC_ORDER_CNT_LSB_MINU→
010- ---- ---- ---- ---- ---- ---- ---- (0x00000009,3) : NUM_REF_FRAMES
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000009,0) : GAPS_IN_FRAME_NUM_VALUE_ALLOWED→
0000 1011 0--- ---- ---- ---- ---- ---- (0x0000000A,7) : PIC_WIDTH_IN_MBS_MINUS1
0001 001- ---- ---- ---- ---- ---- ---- (0x0000000B,6) : PIC_HEIGHT_IN_MAP_UNITS_MINUS1 →
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,7) : FRAME_MBS_ONLY_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,6) : MB_ADAPTIVE_FRAME_FIELD_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,5) : DIRECT_8X8_INFERENCE_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,4) : FRAME_CROPPING_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,3) : VUI_PARAMETERS_PRESENT_FLAG
1--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,2) : RBSP_STOP_ONE_BIT
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,1) : RBSP_ALIGNMENT_ZERO_BIT
```

H.264/AVC example, at start of byte stream (full MTS4EA Trace view window)



```
0000 0000 0000 0000 1000 00-- ---- ---- (0x00000000,7) : PICTURE START CODE
0000 0--- ---- ---- ---- ---- ---- ---- (0x00000002,6) : START_CODE
0000 0000 ---- ---- ---- ---- ---- ---- (0x00000002,1) : TREF
1000 0111 ---- ---- ---- ---- ---- ---- (0x00000003,1) : PTYPE: 1-0-ssi-dci-fpfr-sf
001- ---- ---- ---- ---- ---- ---- ---- (0x00000004,1) : UFEP
0110 1011 1000 1110 00-- ---- ---- ---- (0x00000005,6) : OPPTYPE
0000 0000 1--- ---- ---- ---- ---- ---- (0x00000007,4) : MPPTYPE
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000008,3) : CPM
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000008,2) : UUI
0110 1--- ---- ---- ---- ---- ---- ---- (0x00000008,1) : PQUANT
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000009,4) : PEI
011- ---- ---- ---- ---- ---- ---- ---- (0x00000009,3) : MCBPC_I
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000009,0) : AIC
0110 ---- ---- ---- ---- ---- ---- ---- (0x0000000A,7) : CBPY
0111 ---- ---- ---- ---- ---- ---- ---- (0x0000000A,3) : COEFF
1--- ---- ---- ---- ---- ---- ---- ---- (0x0000000B,7) :       SIGN
0011 00-- ---- ---- ---- ---- ---- ---- (0x0000000B,6) : COEFF
```

H.263 example, at start of bitstream

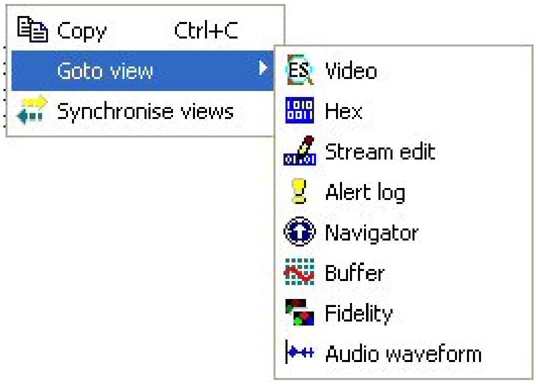
NOTE. *Interpret also provides an excellent tool to do bitstream syntax debugging (see page 6–143).*

To quickly find if there are any errors in the Trace file, use the Find next button function of View trace.., looking for Error or Warning.

Go to other views/right click menu. When you right click, a context-sensitive menu appears (Interpret & Parse Bitstream only) which allows:

- Finding the next instance of the words Error, Warning or Frame, or the entered text

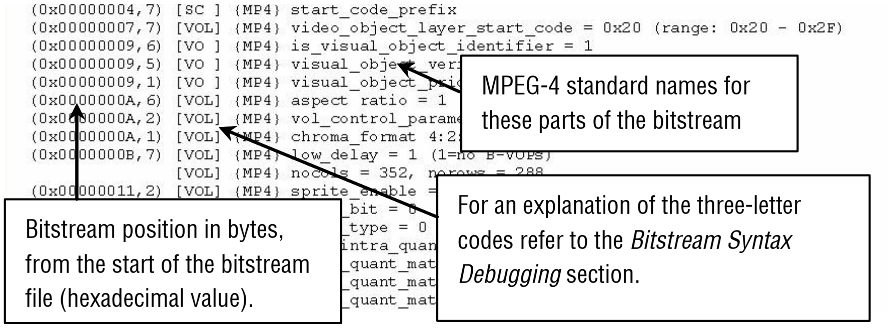
- Going to the area in another view which corresponds to the area that has been selected in the Trace file:



NOTE. *This right-click context-sensitive menu is only active in the Trace/Interpret and Trace/Parse Bitstream trace files.*

Interpret. This provides interpretation of the decoded data in the bitstream. Example outputs are shown in the following diagrams:

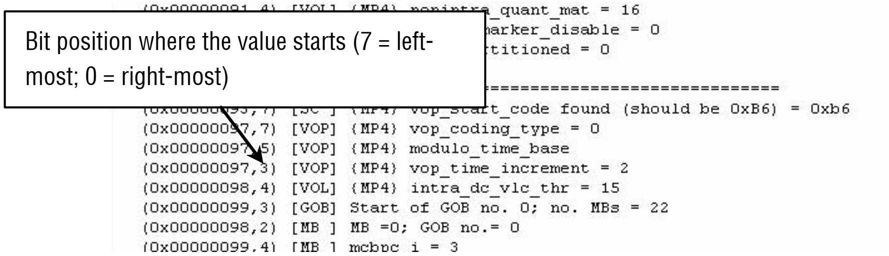
NOTE. *It is important to also see* General codes used in Trace files and Alerts *in the* Bitstream Syntax Debugging *section for an explanation of some of these items.*



```
(0x00000004,7) [SC ] {MP4} start_code_prefix
(0x00000007,7) [VOL] {MP4} video_object_layer_start_code = 0x20 (range: 0x20 - 0x2F)
(0x00000009,6) [VO ] {MP4} is_visual_object_identifier = 1
(0x00000009,5) [VO ] {MP4} visual_object_veri
(0x00000009,1) [VO ] {MP4} visual_object_prio
(0x0000000A,6) [VOL] {MP4} aspect ratio = 1
(0x0000000A,2) [VOL] {MP4} vol_control_parame
(0x0000000A,1) [VOL] {MP4} chroma_format 4:2:
(0x0000000B,7) [VOL] {MP4} low_delay = 1 (1=no B-VOPs)
               [VOL] {MP4} nocols = 352, norows = 288
(0x00000011,2) [VOL] {MP4} sprite_enable =
                                _bit = 0
                                _type = 0
                                intra_quan
                                _quant_mat
                                _quant_mat
                                _quant_mat
```

MPEG-4 standard names for these parts of the bitstream
Bitstream position in bytes, from the start of the bitstream file (hexadecimal value).
For an explanation of the three-letter codes refer to the *Bitstream Syntax Debugging* section.

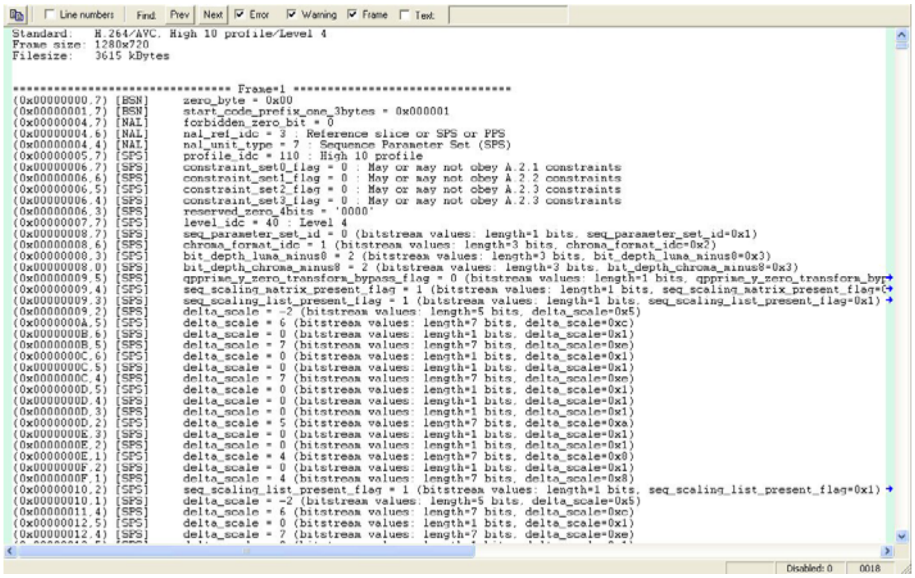
MPEG-4 example, at start of bitstream



```
(0x00000091,4) [VOL] (MP4) nonintra_quant_mat = 16
                           arker_disable = 0
                           titioned = 0

                           ================================
(0x000000  ,7) [SC ] (MP4) vop_start_code found (should be 0xB6) = 0xb6
(0x00000097,7) [VOP] (MP4) vop_coding_type = 0
(0x00000097,5) [VOP] (MP4) modulo_time_base
(0x00000097,3) [VOP] (MP4) vop_time_increment = 2
(0x00000098,4) [VOL] (MP4) intra_dc_vlc_thr = 15
(0x00000099,3) [GOB] Start of GOB no. 0; no. MBs = 22
(0x00000098,2) [MB ] MB =0; GOB no.= 0
(0x00000099,4) [MB ] mcbpc_i = 3
```

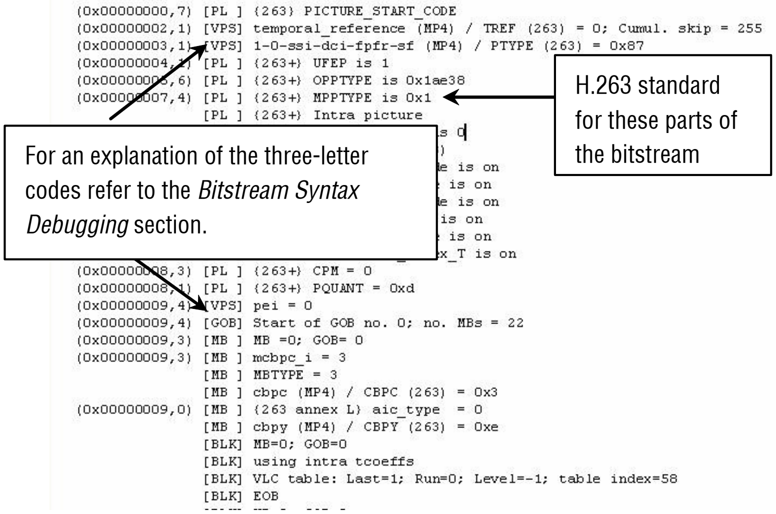
Bit position where the value starts (7 = left-most; 0 = right-most)

MPEG-4 example, at start of VOP



```
Standard:    H.264/AVC, High 10 profile/Level 4
Frame size:  1280x720
Filesize:    3615 kBytes


******************************** Frame=1 ********************************
(0x00000000,7) [BSN]     zero_byte = 0x00
(0x00000001,7) [BSN]     start_code_prefix_one_3bytes = 0x000001
(0x00000004,7) [NAL]     forbidden_zero_bit = 0
(0x00000004,6) [NAL]     nal_ref_idc = 3 : Reference slice or SPS or PPS
(0x00000004,4) [NAL]     nal_unit_type = 7 : Sequence Parameter Set (SPS)
(0x00000005,7) [SPS]     profile_idc = 110 : High 10 profile
(0x00000006,7) [SPS]     constraint_set0_flag = 0 : May or may not obey A.2.1 constraints
(0x00000006,6) [SPS]     constraint_set1_flag = 0 : May or may not obey A.2.2 constraints
(0x00000006,5) [SPS]     constraint_set2_flag = 0 : May or may not obey A.2.3 constraints
(0x00000006,4) [SPS]     constraint_set3_flag = 0 : May or may not obey A.2.3 constraints
(0x00000006,3) [SPS]     reserved_zero_4bits = '0000'
(0x00000007,7) [SPS]     level_idc = 40 : Level 4
(0x00000008,7) [SPS]     seq_parameter_set_id = 0 (bitstream values: length=1 bits, seq_parameter_set_id=0x1)
(0x00000008,6) [SPS]     chroma_format_idc = 1 (bitstream values: length=3 bits, chroma_format_idc=0x2)
(0x00000008,3) [SPS]     bit_depth_luma_minus8 = 2 (bitstream values: length=3 bits, bit_depth_luma_minus8=0x3)
(0x00000008,0) [SPS]     bit_depth_chroma_minus8 = 2 (bitstream values: length=3 bits, bit_depth_chroma_minus8=0x3)
(0x00000009,5) [SPS]     qpprime_y_zero_transform_bypass_flag = 0 (bitstream values: length=1 bits, qpprime_y_zero_transform_byp→
(0x00000009,4) [SPS]     seq_scaling_matrix_present_flag = 1 (bitstream values: length=1 bits, seq_scaling_matrix_present_flag=(→
(0x00000009,3) [SPS]     seq_scaling_list_present_flag = 1 (bitstream values: length=1 bits, seq_scaling_list_present_flag=0x1) →
(0x00000009,2) [SPS]     delta_scale = -2 (bitstream values: length=5 bits, delta_scale=0x5)
(0x0000000A,5) [SPS]     delta_scale = 6 (bitstream values: length=7 bits, delta_scale=0xc)
(0x0000000B,6) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x0000000B,5) [SPS]     delta_scale = 7 (bitstream values: length=7 bits, delta_scale=0xe)
(0x0000000C,6) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x0000000C,5) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x0000000C,4) [SPS]     delta_scale = 7 (bitstream values: length=7 bits, delta_scale=0xe)
(0x0000000D,5) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x0000000D,4) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x0000000D,3) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x0000000D,2) [SPS]     delta_scale = 5 (bitstream values: length=7 bits, delta_scale=0xa)
(0x0000000E,3) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x0000000E,2) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x0000000E,1) [SPS]     delta_scale = 4 (bitstream values: length=7 bits, delta_scale=0x8)
(0x0000000F,2) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x0000000F,1) [SPS]     delta_scale = 4 (bitstream values: length=7 bits, delta_scale=0x8)
(0x00000010,2) [SPS]     seq_scaling_list_present_flag = 1 (bitstream values: length=1 bits, seq_scaling_list_present_flag=0x1) →
(0x00000010,1) [SPS]     delta_scale = -2 (bitstream values: length=5 bits, delta_scale=0x5)
(0x00000011,4) [SPS]     delta_scale = 6 (bitstream values: length=7 bits, delta_scale=0xc)
(0x00000012,5) [SPS]     delta_scale = 0 (bitstream values: length=1 bits, delta_scale=0x1)
(0x00000012,4) [SPS]     delta_scale = 7 (bitstream values: length=7 bits, delta_scale=0xe)
```

H.264/AVC example, at start of byte stream (full MTS4EA Trace view window)



```
(0x00000000,7) [PL ] (263) PICTURE_START_CODE
(0x00000002,1) [VPS] temporal_reference (MP4) / TREF (263) = 0; Cumul. skip = 255
(0x00000003,1) [VPS] 1-0-ssi-dci-fpfr-sf (MP4) / PTYPE (263) = 0x87
(0x00000004,1) [PL ] (263+) UFEP is 1
(0x00000005,6) [PL ] (263+) OPPTYPE is 0x1ae38
(0x00000007,4) [PL ] (263+) MPPTYPE is 0x1
               [PL ] (263+) Intra picture
                                             s 0
                                             )
                                             e is on
                                             is on
                                             e is on
                                             is on
                                             is on
                                             x_T is on
(0x00000008,3) [PL ] (263+) CPM = 0
(0x00000008,1) [PL ] (263+) PQUANT = 0xd
(0x00000009,4) [VPS] pei = 0
(0x00000009,4) [GOB] Start of GOB no. 0; no. MBs = 22
(0x00000009,3) [MB ] MB =0; GOB= 0
(0x00000009,3) [MB ] mcbpc_i = 3
               [MB ] MBTYPE = 3
               [MB ] cbpc (MP4) / CBPC (263) = 0x3
(0x00000009,0) [MB ] (263 annex L) aic_type  = 0
               [MB ] cbpy (MP4) / CBPY (263) = 0xe
               [BLK] MB=0; GOB=0
               [BLK] using intra tcoeffs
               [BLK] VLC table: Last=1; Run=0; Level=-1; table index=58
               [BLK] EOB
```

H.263 standard for these parts of the bitstream

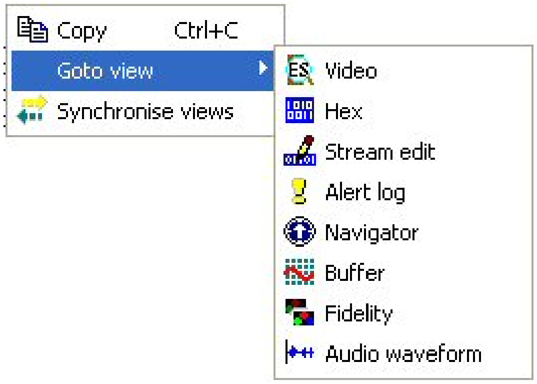
For an explanation of the three-letter codes refer to the *Bitstream Syntax Debugging* section.

H.263 Example at start of bitstream.

NOTE. *To quickly find if there are any errors in the Trace file, use the Find next function of View trace..., looking for the Error and Warning.*

When you right-click, a context-sensitive menu appears (Interpret and Parse Bitstream only), which allows you to:

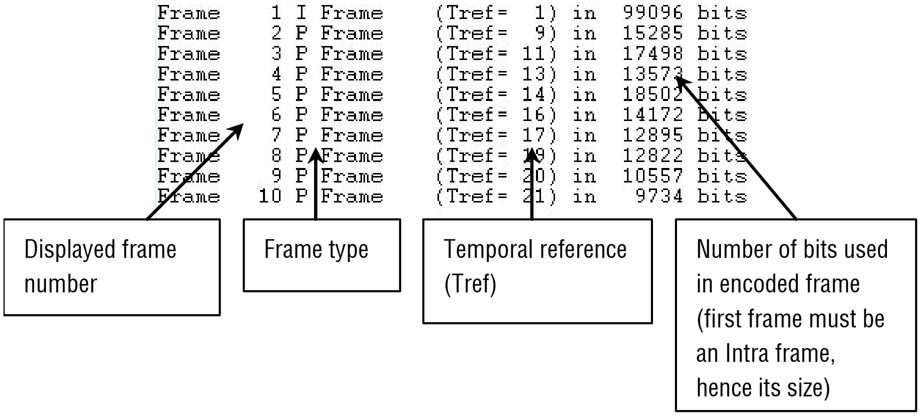
- Find the next instance of the words Error, Warning or Frame, or the entered text
- Go to the area in another view that corresponds to the area selected in the Trace file:



NOTE. *This right-click context-sensitive menu is only active in the Trace/ Interpret and Trace/Parse Bitstream trace files.*

Frame Summary [H.264/AVC, MPEG 2, and H.263] or **VOP Summary** [MPEG 4] If this level of Trace is selected, then the Frame summary is output to the file, specifying the number of bits generated, the index of the decoded frame, the frame type, the temporal reference (if appropriate) and other data.

This example is for H.263, where Tref is used.



```
Frame    1 I Frame    (Tref=  1) in  99096 bits
Frame    2 P Frame    (Tref=  9) in  15285 bits
Frame    3 P Frame    (Tref= 11) in  17498 bits
Frame    4 P Frame    (Tref= 13) in  13573 bits
Frame    5 P Frame    (Tref= 14) in  18502 bits
Frame    6 P Frame    (Tref= 16) in  14172 bits
Frame    7 P Frame    (Tref= 17) in  12895 bits
Frame    8 P Frame    (Tref= 19) in  12822 bits
Frame    9 P Frame    (Tref= 20) in  10557 bits
Frame   10 P Frame    (Tref= 21) in   9734 bits
```

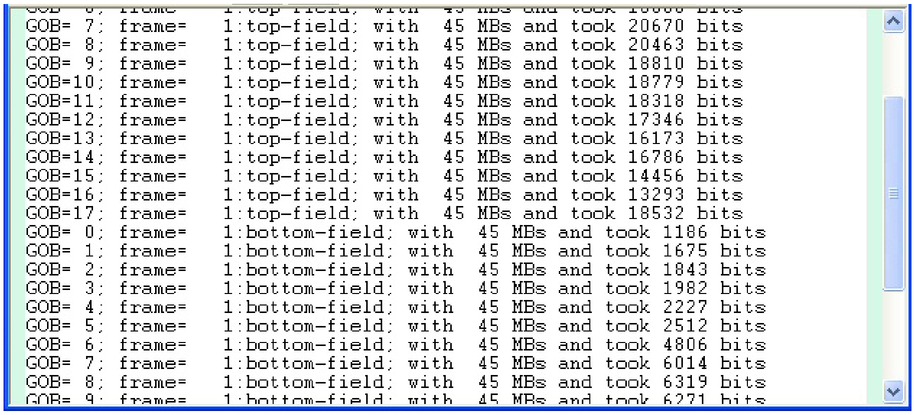
Displayed frame number

Frame type

Temporal reference (Tref)

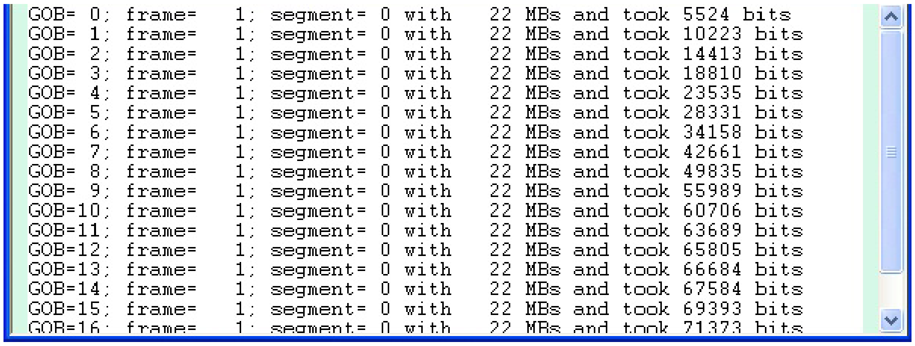
Number of bits used in encoded frame (first frame must be an Intra frame, hence its size)

MB row summary [H.264/AVC, MPEG 4, MPEG 2] or **GOB Summary** [H.263]. This Trace outputs summary data for each horizontal row of MacroBlocks, starting from the top of the frame and working downwards.



```
GOB= 7; frame=   1:top-field; with  45 MBs and took 20670 bits
GOB= 8; frame=   1:top-field; with  45 MBs and took 20463 bits
GOB= 9; frame=   1:top-field; with  45 MBs and took 18810 bits
GOB=10; frame=   1:top-field; with  45 MBs and took 18779 bits
GOB=11; frame=   1:top-field; with  45 MBs and took 18318 bits
GOB=12; frame=   1:top-field; with  45 MBs and took 17346 bits
GOB=13; frame=   1:top-field; with  45 MBs and took 16173 bits
GOB=14; frame=   1:top-field; with  45 MBs and took 16786 bits
GOB=15; frame=   1:top-field; with  45 MBs and took 14456 bits
GOB=16; frame=   1:top-field; with  45 MBs and took 13293 bits
GOB=17; frame=   1:top-field; with  45 MBs and took 18532 bits
GOB= 0; frame=   1:bottom-field; with  45 MBs and took 1186 bits
GOB= 1; frame=   1:bottom-field; with  45 MBs and took 1675 bits
GOB= 2; frame=   1:bottom-field; with  45 MBs and took 1843 bits
GOB= 3; frame=   1:bottom-field; with  45 MBs and took 1982 bits
GOB= 4; frame=   1:bottom-field; with  45 MBs and took 2227 bits
GOB= 5; frame=   1:bottom-field; with  45 MBs and took 2512 bits
GOB= 6; frame=   1:bottom-field; with  45 MBs and took 4806 bits
GOB= 7; frame=   1:bottom-field; with  45 MBs and took 6014 bits
GOB= 8; frame=   1:bottom-field; with  45 MBs and took 6319 bits
GOB= 9; frame=   1:bottom-field; with  45 MBs and took 6271 bits
```

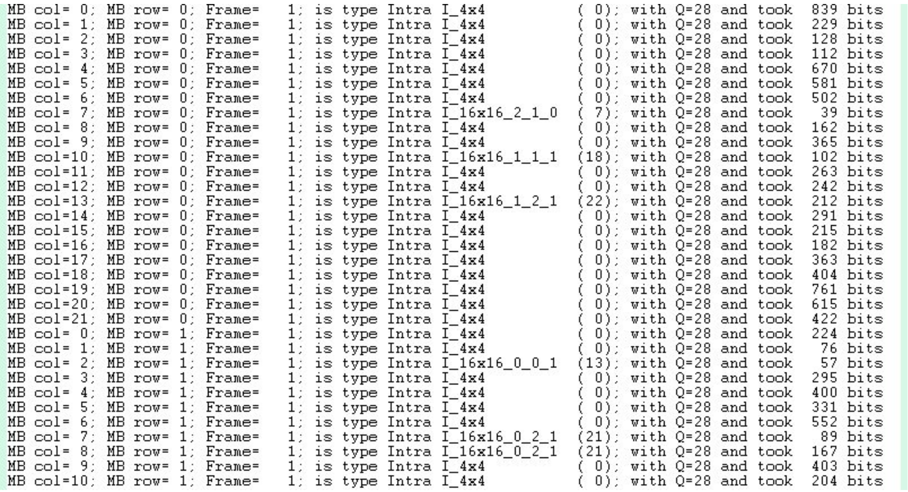
MPEG-2 example (interlaced example with top and bottom fields)



```
GOB= 0; frame=   1; segment= 0 with   22 MBs and took 5524 bits
GOB= 1; frame=   1; segment= 0 with   22 MBs and took 10223 bits
GOB= 2; frame=   1; segment= 0 with   22 MBs and took 14413 bits
GOB= 3; frame=   1; segment= 0 with   22 MBs and took 18810 bits
GOB= 4; frame=   1; segment= 0 with   22 MBs and took 23535 bits
GOB= 5; frame=   1; segment= 0 with   22 MBs and took 28331 bits
GOB= 6; frame=   1; segment= 0 with   22 MBs and took 34158 bits
GOB= 7; frame=   1; segment= 0 with   22 MBs and took 42661 bits
GOB= 8; frame=   1; segment= 0 with   22 MBs and took 49835 bits
GOB= 9; frame=   1; segment= 0 with   22 MBs and took 55989 bits
GOB=10; frame=   1; segment= 0 with   22 MBs and took 60706 bits
GOB=11; frame=   1; segment= 0 with   22 MBs and took 63689 bits
GOB=12; frame=   1; segment= 0 with   22 MBs and took 65805 bits
GOB=13; frame=   1; segment= 0 with   22 MBs and took 66684 bits
GOB=14; frame=   1; segment= 0 with   22 MBs and took 67584 bits
GOB=15; frame=   1; segment= 0 with   22 MBs and took 69393 bits
GOB=16; frame=   1; segment= 0 with   22 MBs and took 71373 bits
```

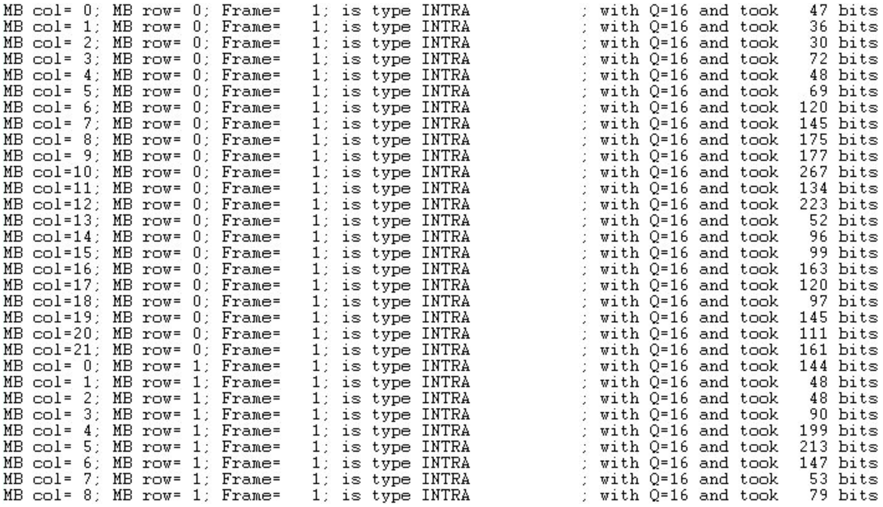
H.263 example
(see *Segment* [H.263 only] on page 6–88)

MB Summary. For each MacroBlock, this option provides the location of the MacroBlock (MacroBlock column and row, frame), the type of coding used, the quantizer used, and the number of bits used. MB col=0, MB row=0 is the top left MacroBlock in the frame.



```
MB col= 0; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  839 bits
MB col= 1; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  229 bits
MB col= 2; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  128 bits
MB col= 3; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  112 bits
MB col= 4; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  670 bits
MB col= 5; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  581 bits
MB col= 6; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  502 bits
MB col= 7; MB row= 0; Frame=   1; is type Intra I_16x16_2_1_0  ( 7); with Q=28 and took   39 bits
MB col= 8; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  162 bits
MB col= 9; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  365 bits
MB col=10; MB row= 0; Frame=   1; is type Intra I_16x16_1_1_1  (18); with Q=28 and took  102 bits
MB col=11; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  263 bits
MB col=12; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  242 bits
MB col=13; MB row= 0; Frame=   1; is type Intra I_16x16_1_2_1  (22); with Q=28 and took  212 bits
MB col=14; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  291 bits
MB col=15; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  215 bits
MB col=16; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  182 bits
MB col=17; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  363 bits
MB col=18; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  404 bits
MB col=19; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  761 bits
MB col=20; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  615 bits
MB col=21; MB row= 0; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  422 bits
MB col= 0; MB row= 1; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  224 bits
MB col= 1; MB row= 1; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took   76 bits
MB col= 2; MB row= 1; Frame=   1; is type Intra I_16x16_0_0_1  (13); with Q=28 and took   57 bits
MB col= 3; MB row= 1; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  295 bits
MB col= 4; MB row= 1; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  400 bits
MB col= 5; MB row= 1; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  331 bits
MB col= 6; MB row= 1; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  552 bits
MB col= 7; MB row= 1; Frame=   1; is type Intra I_16x16_0_2_1  (21); with Q=28 and took   89 bits
MB col= 8; MB row= 1; Frame=   1; is type Intra I_16x16_0_2_1  (21); with Q=28 and took  167 bits
MB col= 9; MB row= 1; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  403 bits
MB col=10; MB row= 1; Frame=   1; is type Intra I_4x4          ( 0); with Q=28 and took  204 bits
```

H.264/AVC example



```
MB col= 0; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   47 bits
MB col= 1; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   36 bits
MB col= 2; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   30 bits
MB col= 3; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   72 bits
MB col= 4; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   48 bits
MB col= 5; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   69 bits
MB col= 6; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  120 bits
MB col= 7; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  145 bits
MB col= 8; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  175 bits
MB col= 9; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  177 bits
MB col=10; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  267 bits
MB col=11; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  134 bits
MB col=12; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  223 bits
MB col=13; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   52 bits
MB col=14; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   96 bits
MB col=15; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   99 bits
MB col=16; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  163 bits
MB col=17; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  120 bits
MB col=18; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took   97 bits
MB col=19; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  145 bits
MB col=20; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  111 bits
MB col=21; MB row= 0; Frame=   1; is type INTRA          ; with Q=16 and took  161 bits
MB col= 0; MB row= 1; Frame=   1; is type INTRA          ; with Q=16 and took  144 bits
MB col= 1; MB row= 1; Frame=   1; is type INTRA          ; with Q=16 and took   48 bits
MB col= 2; MB row= 1; Frame=   1; is type INTRA          ; with Q=16 and took   48 bits
MB col= 3; MB row= 1; Frame=   1; is type INTRA          ; with Q=16 and took   90 bits
MB col= 4; MB row= 1; Frame=   1; is type INTRA          ; with Q=16 and took  199 bits
MB col= 5; MB row= 1; Frame=   1; is type INTRA          ; with Q=16 and took  213 bits
MB col= 6; MB row= 1; Frame=   1; is type INTRA          ; with Q=16 and took  147 bits
MB col= 7; MB row= 1; Frame=   1; is type INTRA          ; with Q=16 and took   53 bits
MB col= 8; MB row= 1; Frame=   1; is type INTRA          ; with Q=16 and took   79 bits
```

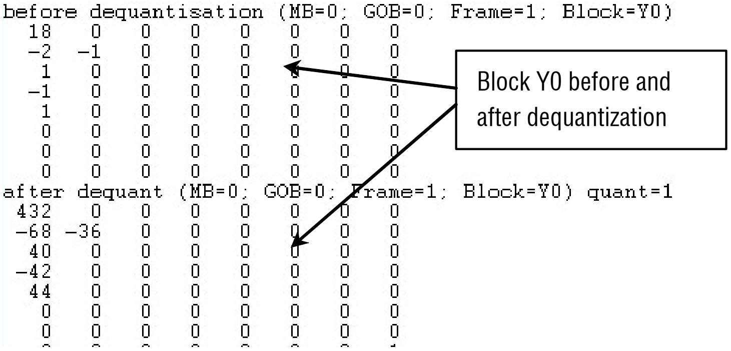
MPEG-4 example

Transform Level [H.264/AVC] or **DCT Level** [MPEG 4, MPEG 2, and H.263].

NOTE. *In H.264/AVC the report is of transform coefficients that are not DCTs.*

For MPEG-4, and H263 this gives three sets of information:

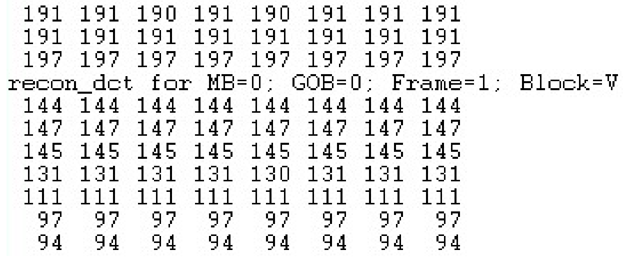
- Before dequantization = the values of the DCT coefficients in the bitstream
- After dequantization = the before values after they have been multiplied by the quantizer matrix



```
before dequantisation (MB=0; GOB=0; Frame=1; Block=Y0)
  18   0   0   0   0   0   0   0
  -2  -1   0   0   0   0   0   0
   1   0   0   0   0   0   0   0
  -1   0   0   0   0   0   0   0
   1   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0
after dequant (MB=0; GOB=0; Frame=1; Block=Y0) quant=1
 432   0   0   0   0   0   0   0
 -68 -36   0   0   0   0   0   0
  40   0   0   0   0   0   0   0
 -42   0   0   0   0   0   0   0
  44   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0
```

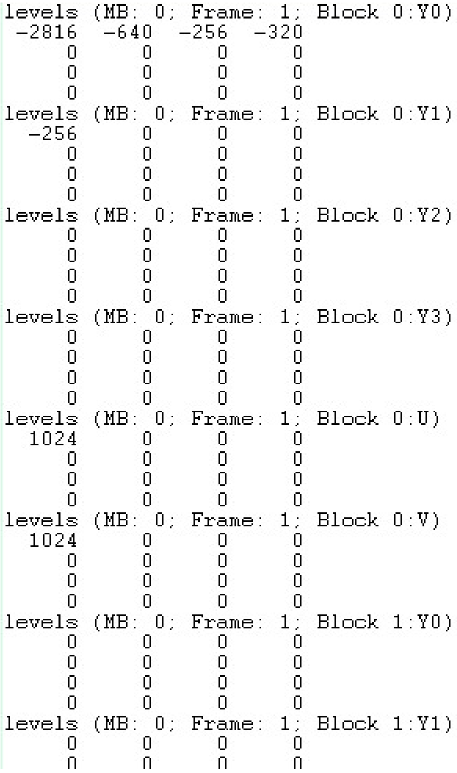
Block Y0 before and after dequantization

- `recon_dct` = the inverse DCT residual values, before summing with any prediction. The data is written into the file in two separate places - the second part is below:



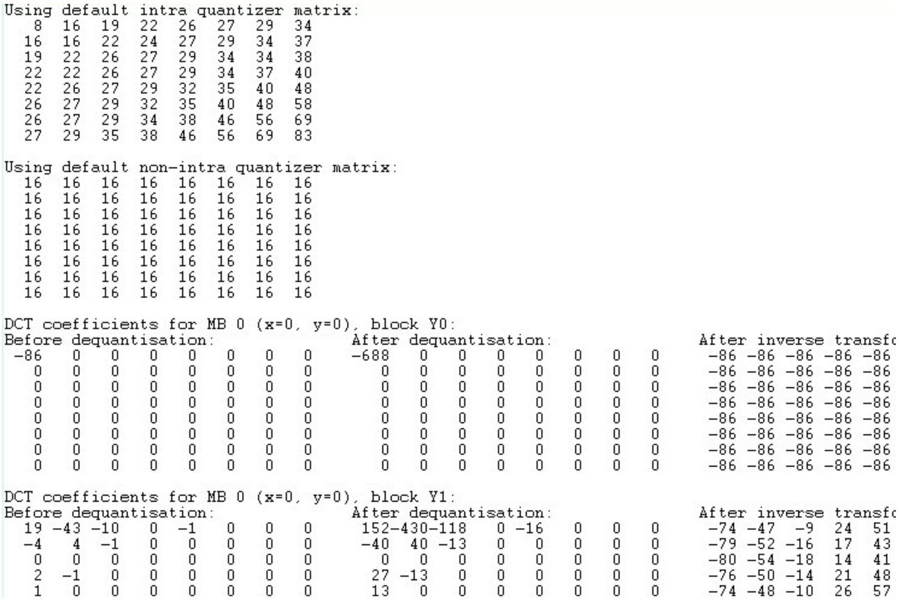
```
 191 191 190 191 190 191 191 191
 191 191 191 191 191 191 191 191
 197 197 197 197 197 197 197 197
recon_dct for MB=0; GOB=0; Frame=1; Block=V
 144 144 144 144 144 144 144 144
 147 147 147 147 147 147 147 147
 145 145 145 145 145 145 145 145
 131 131 131 131 130 131 131 131
 111 111 111 111 111 111 111 111
  97  97  97  97  97  97  97  97
  94  94  94  94  94  94  94  94
```

In H.264/AVC the report is of transform levels:



```
levels (MB: 0; Frame: 1; Block 0:Y0)
 -2816  -640  -256  -320
     0     0     0     0
     0     0     0     0
     0     0     0     0
levels (MB: 0; Frame: 1; Block 0:Y1)
  -256     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
levels (MB: 0; Frame: 1; Block 0:Y2)
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
levels (MB: 0; Frame: 1; Block 0:Y3)
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
levels (MB: 0; Frame: 1; Block 0:U)
  1024     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
levels (MB: 0; Frame: 1; Block 0:V)
  1024     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
levels (MB: 0; Frame: 1; Block 1:Y0)
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
levels (MB: 0; Frame: 1; Block 1:Y1)
     0     0     0     0
     0     0     0     0
```

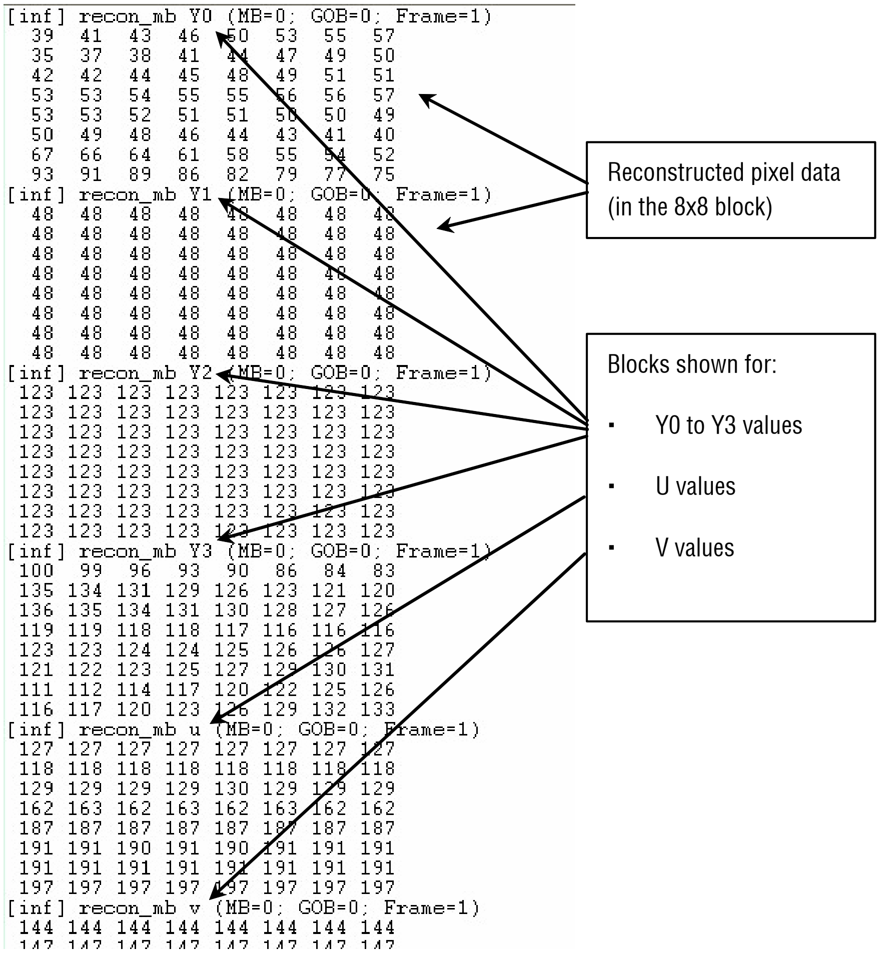
In MPEG-2 the data is reported in the following format:



```
Using default intra quantizer matrix:
   8  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

Using default non-intra quantizer matrix:
  16  16  16  16  16  16  16  16
  16  16  16  16  16  16  16  16
  16  16  16  16  16  16  16  16
  16  16  16  16  16  16  16  16
  16  16  16  16  16  16  16  16
  16  16  16  16  16  16  16  16
  16  16  16  16  16  16  16  16
  16  16  16  16  16  16  16  16

DCT coefficients for MB 0 (x=0, y=0), block Y0:
Before dequantisation:                  After dequantisation:                   After inverse transfc
 -86   0   0   0   0   0   0   0    -688   0   0   0   0   0   0   0     -86 -86 -86 -86 -86
   0   0   0   0   0   0   0   0       0   0   0   0   0   0   0   0     -86 -86 -86 -86 -86
   0   0   0   0   0   0   0   0       0   0   0   0   0   0   0   0     -86 -86 -86 -86 -86
   0   0   0   0   0   0   0   0       0   0   0   0   0   0   0   0     -86 -86 -86 -86 -86
   0   0   0   0   0   0   0   0       0   0   0   0   0   0   0   0     -86 -86 -86 -86 -86
   0   0   0   0   0   0   0   0       0   0   0   0   0   0   0   0     -86 -86 -86 -86 -86
   0   0   0   0   0   0   0   0       0   0   0   0   0   0   0   0     -86 -86 -86 -86 -86
   0   0   0   0   0   0   0   0       0   0   0   0   0   0   0   0     -86 -86 -86 -86 -86

DCT coefficients for MB 0 (x=0, y=0), block Y1:
Before dequantisation:                  After dequantisation:                   After inverse transfc
  19 -43 -10   0  -1   0   0   0     152-430-118   0 -16   0   0   0     -74 -47  -9  24  51
  -4   4  -1   0   0   0   0   0     -40  40 -13   0   0   0   0   0     -79 -52 -16  17  43
   0   0   0   0   0   0   0   0       0   0   0   0   0   0   0   0     -80 -54 -18  14  41
   2  -1   0   0   0   0   0   0      27 -13   0   0   0   0   0   0     -76 -50 -14  21  48
   1   0   0   0   0   0   0   0      13   0   0   0   0   0   0   0     -74 -48 -10  26  57
```

NOTE. *This can generate a lot of data.*

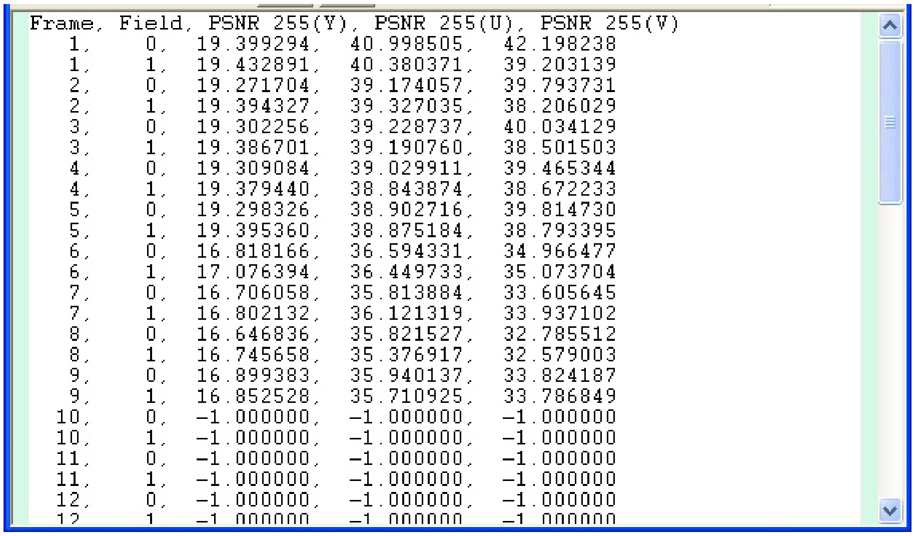
Pixel level. This lowest level of Trace provides the output pixel values in 8x8 blocks for the video decoding process.



```
[inf] recon_mb Y0 (MB=0; GOB=0; Frame=1)
  39  41  43  46  50  53  55  57
  35  37  38  41  44  47  49  50
  42  42  44  45  48  49  51  51
  53  53  54  55  55  56  56  57
  53  53  52  51  51  50  50  49
  50  49  48  46  44  43  41  40
  67  66  64  61  58  55  54  52
  93  91  89  86  82  79  77  75
[inf] recon_mb Y1 (MB=0; GOB=0; Frame=1)
  48  48  48  48  48  48  48  48
  48  48  48  48  48  48  48  48
  48  48  48  48  48  48  48  48
  48  48  48  48  48  48  48  48
  48  48  48  48  48  48  48  48
  48  48  48  48  48  48  48  48
  48  48  48  48  48  48  48  48
  48  48  48  48  48  48  48  48
[inf] recon_mb Y2 (MB=0; GOB=0; Frame=1)
 123 123 123 123 123 123 123 123
 123 123 123 123 123 123 123 123
 123 123 123 123 123 123 123 123
 123 123 123 123 123 123 123 123
 123 123 123 123 123 123 123 123
 123 123 123 123 123 123 123 123
 123 123 123 123 123 123 123 123
 123 123 123 123 123 123 123 123
[inf] recon_mb Y3 (MB=0; GOB=0; Frame=1)
 100  99  96  93  90  86  84  83
 135 134 131 129 126 123 121 120
 136 135 134 131 130 128 127 126
 119 119 118 118 117 116 116 116
 123 123 124 124 125 126 126 127
 121 122 123 125 127 129 130 131
 111 112 114 117 120 122 125 126
 116 117 120 123 126 129 132 133
[inf] recon_mb u (MB=0; GOB=0; Frame=1)
 127 127 127 127 127 127 127 127
 118 118 118 118 118 118 118 118
 129 129 129 129 130 129 129 129
 162 163 162 163 162 163 162 162
 187 187 187 187 187 187 187 187
 191 191 190 191 190 191 191 191
 191 191 191 191 191 191 191 191
 197 197 197 197 197 197 197 197
[inf] recon_mb v (MB=0; GOB=0; Frame=1)
 144 144 144 144 144 144 144 144
 147 147 147 147 147 147 147 147
```

NOTE. *This can generate a lot of data.*

Frame fidelity [H.264/AVC, MPEG 2, and H.263] or **VOP fidelity** [MPEG 4]. This provides a Trace output of the fidelity analysis frame-by-frame. See Video Fidelity Enable on page 6–162 for more information on fidelity analysis.

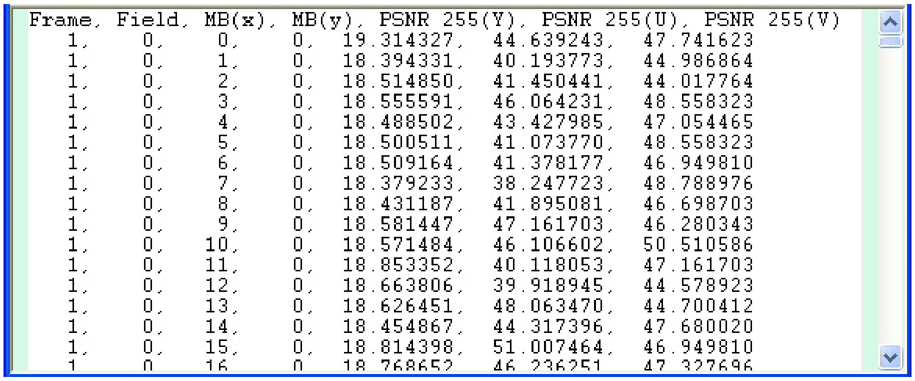


```
Frame, Field, PSNR 255(Y), PSNR 255(U), PSNR 255(V)
    1,     0,  19.399294,   40.998505,   42.198238
    1,     1,  19.432891,   40.380371,   39.203139
    2,     0,  19.271704,   39.174057,   39.793731
    2,     1,  19.394327,   39.327035,   38.206029
    3,     0,  19.302256,   39.228737,   40.034129
    3,     1,  19.386701,   39.190760,   38.501503
    4,     0,  19.309084,   39.029911,   39.465344
    4,     1,  19.379440,   38.843874,   38.672233
    5,     0,  19.298326,   38.902716,   39.814730
    5,     1,  19.395360,   38.875184,   38.793395
    6,     0,  16.818166,   36.594331,   34.966477
    6,     1,  17.076394,   36.449733,   35.073704
    7,     0,  16.706058,   35.813884,   33.605645
    7,     1,  16.802132,   36.121319,   33.937102
    8,     0,  16.646836,   35.821527,   32.785512
    8,     1,  16.745658,   35.376917,   32.579003
    9,     0,  16.899383,   35.940137,   33.824187
    9,     1,  16.852528,   35.710925,   33.786849
   10,     0,  -1.000000,  -1.000000,  -1.000000
   10,     1,  -1.000000,  -1.000000,  -1.000000
   11,     0,  -1.000000,  -1.000000,  -1.000000
   11,     1,  -1.000000,  -1.000000,  -1.000000
   12,     0,  -1.000000,  -1.000000,  -1.000000
   12,     1,  -1.000000,  -1.000000,  -1.000000
```

NOTE. *The lines in the preceding screenshot for frames 10 and above are empty because the corresponding YUV file stops at this point (there are no more frames in the YUV file). The -1 in the trace file indicates that the YUV file is missing.*

Some of the example files provided have the necessary corresponding YUV files for fidelity analysis - see YUV source files for Example Fidelity Analysis *on page 6–43.*

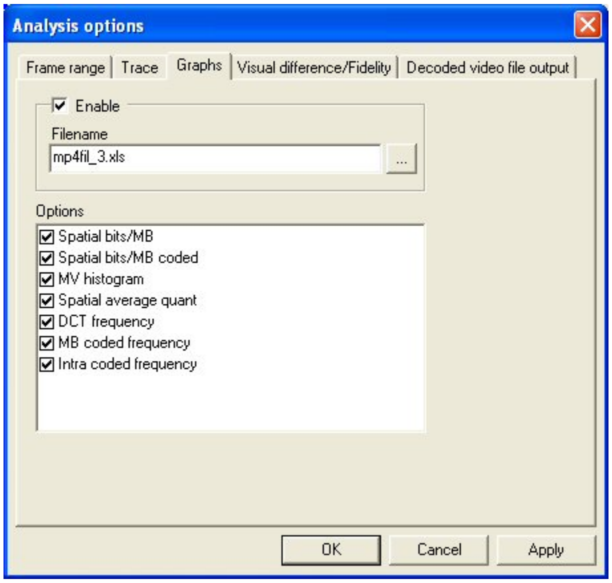
MacroBlock fidelity. This provides a Trace output of the fidelity analysis for each MacroBlock in the frame range. See Video Fidelity Enable on page 6–162 for more information on fidelity analysis.



```
Frame, Field, MB(x), MB(y), PSNR 255(Y), PSNR 255(U), PSNR 255(V)
    1,     0,     0,     0,  19.314327,   44.639243,   47.741623
    1,     0,     1,     0,  18.394331,   40.193773,   44.986864
    1,     0,     2,     0,  18.514850,   41.450441,   44.017764
    1,     0,     3,     0,  18.555591,   46.064231,   48.558323
    1,     0,     4,     0,  18.488502,   43.427985,   47.054465
    1,     0,     5,     0,  18.500511,   41.073770,   48.558323
    1,     0,     6,     0,  18.509164,   41.378177,   46.949810
    1,     0,     7,     0,  18.379233,   38.247723,   48.788976
    1,     0,     8,     0,  18.431187,   41.895081,   46.698703
    1,     0,     9,     0,  18.581447,   47.161703,   46.280343
    1,     0,    10,     0,  18.571484,   46.106602,   50.510586
    1,     0,    11,     0,  18.853352,   40.118053,   47.161703
    1,     0,    12,     0,  18.663806,   39.918945,   44.578923
    1,     0,    13,     0,  18.626451,   48.063470,   44.700412
    1,     0,    14,     0,  18.454867,   44.317396,   47.680020
    1,     0,    15,     0,  18.814398,   51.007464,   46.949810
    1,     0,    16,     0,  18.768652,   46.236251,   47.327696
```

NOTE. *Some of the example files provided have the necessary corresponding YUV files for fidelity analysis - see see* YUV source files for Example Fidelity Analysis *on page 6–43.*

Video Graph Enable Ctrl+G



☑ Enable

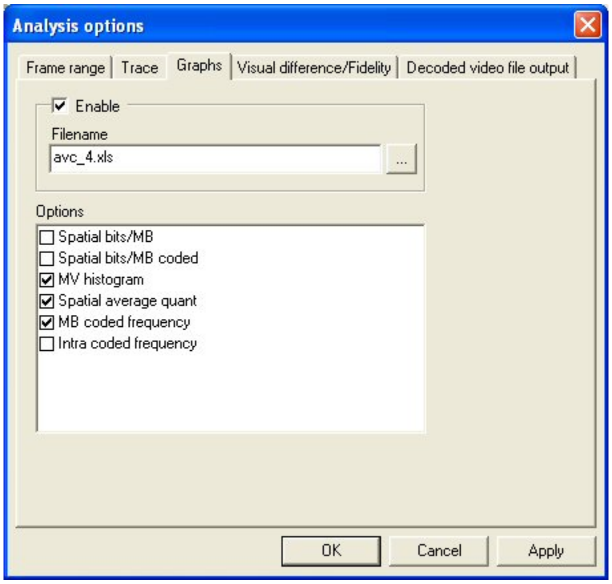
Filename: mp4fil_3.xls

Options

- ☑ Spatial bits/MB
- ☑ Spatial bits/MB coded
- ☑ MV histogram
- ☑ Spatial average quant
- ☑ DCT frequency
- ☑ MB coded frequency
- ☑ Intra coded frequency

Graph enable... turns on the collection of statistics for display using Microsoft Excel. It does this by saving the statistics of the decoding process in an `.xls` file. Microsoft Excel is then run and a macro converts the statistics data into graphs, which are then available in various tabs of Excel.

The collection of statistics varies depending on the standard concerned - for example, there is no DCT as such in H.264/AVC, so this is not available:



☑ Enable

Filename: avc_4.xls

Options

- ☐ Spatial bits/MB
- ☐ Spatial bits/MB coded
- ☑ MV histogram
- ☑ Spatial average quant
- ☑ MB coded frequency
- ☐ Intra coded frequency

Many of these graphs are equivalent to the statistical summaries that can be overlaid on the final frame, but are presented in a more visual format.

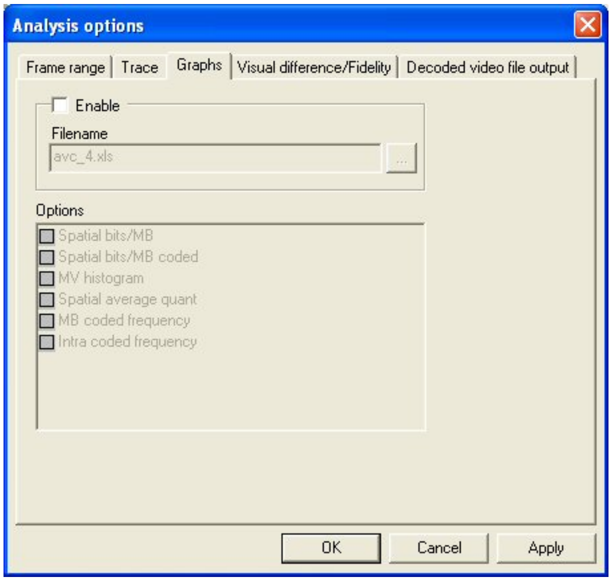
NOTE. *The MTS4EA assumes that the Microsoft Excel program is used for analysis of the provided statistics, and the macro given as part of MTS4EA is designed to operate with Excel versions 97, 2000 and XP.*

Frame Range Tab

You can set the range of frames over which the statistics are collected can be set.

For details on how to do this, see *Frame Range Tab* on page 6–137.

Enable. This option enables/disables the collection of the statistics for the Graph data. When disabled, the Graph tab appears as follows:



□ Enable

Filename

avc_4.xls

Options

□ Spatial bits/MB
□ Spatial bits/MB coded
□ MV histogram
□ Spatial average quant
□ MB coded frequency
□ Intra coded frequency

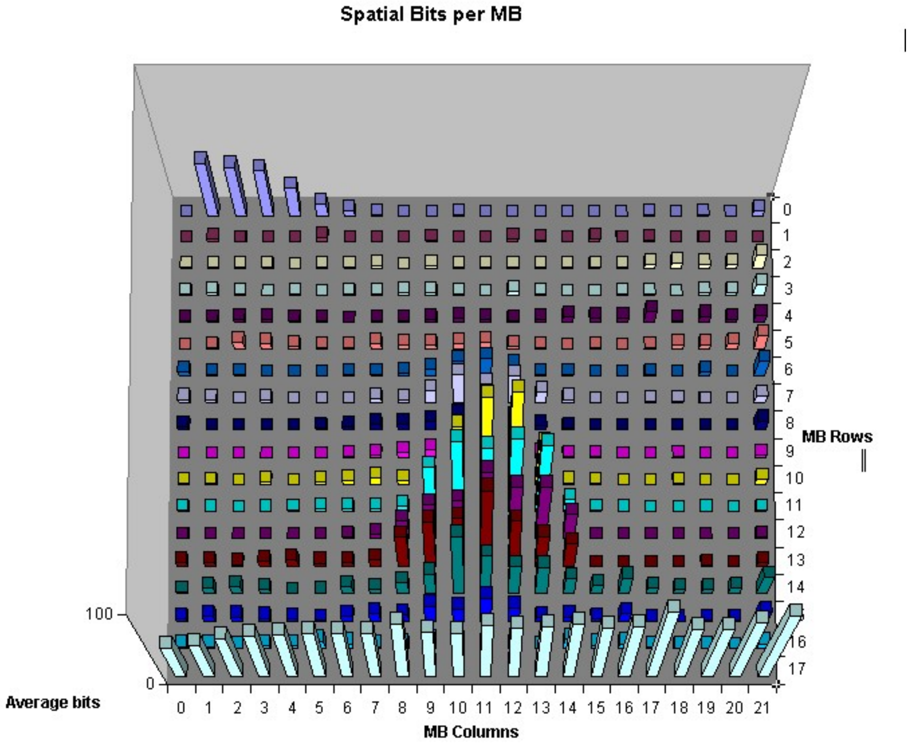
Filename. By default, the suggested filename is the same as the base name of the input video file given the extension .xls.

A different filename can be entered (but not a different file extension), and the folder where the file is to be stored can be changed by clicking the browse box.

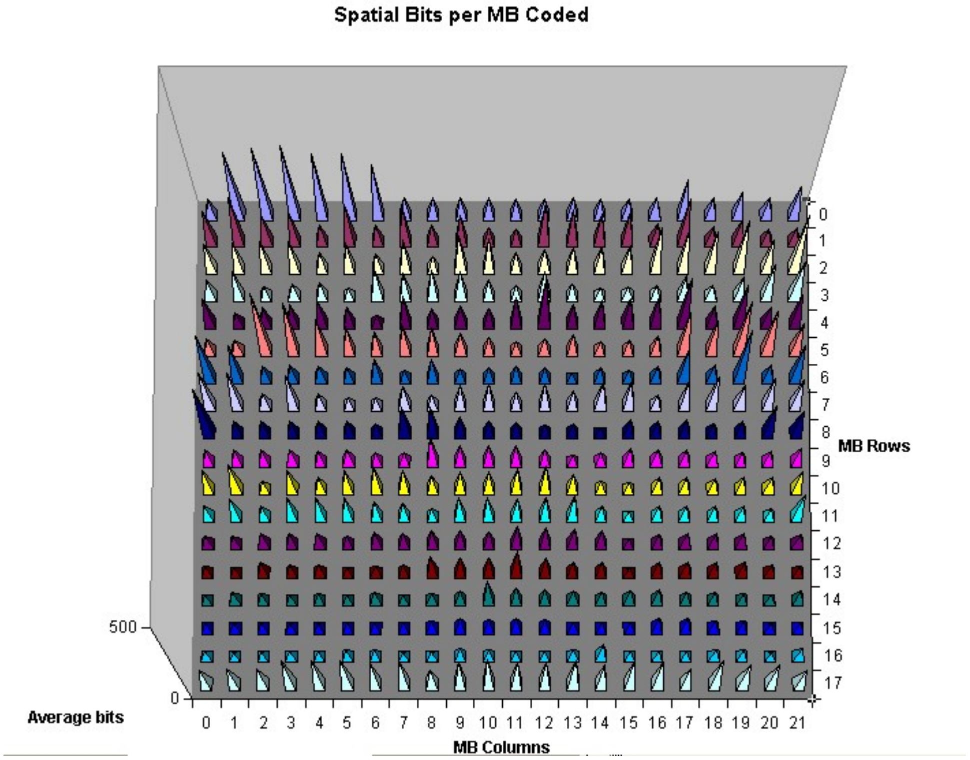
NOTE. *An Excel spreadsheet can only be 256 columns wide - if the motion vectors are outside the range ± 127, the values are put into bins to scale the values within the 256 columns.*

Graphs Tab

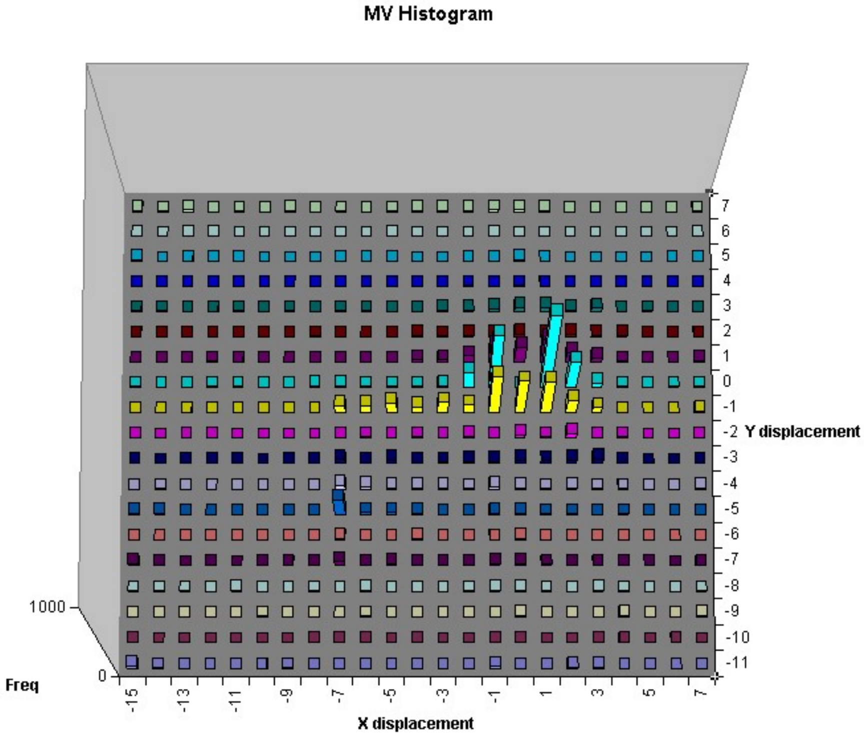
Spatial bits/MB. This option creates a graph that shows the average bits per MacroBlock as a contour plot, over a two-dimensional plane that represents the picture. The contours are plotted in different colors, and the meanings of the colors are represented in a key positioned at the side of the graph.



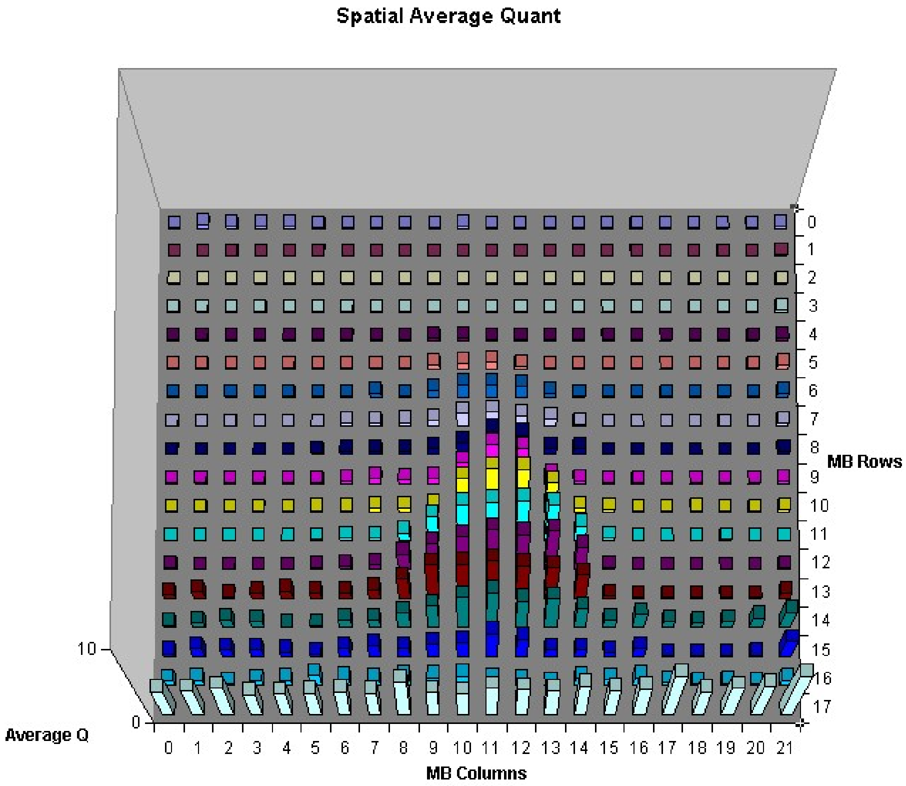
Spatial bits/MB coded. This option creates a three-dimensional pyramidal graph, where the height of the pyramid represents the average number of bits each MacroBlock used for each time it was coded.



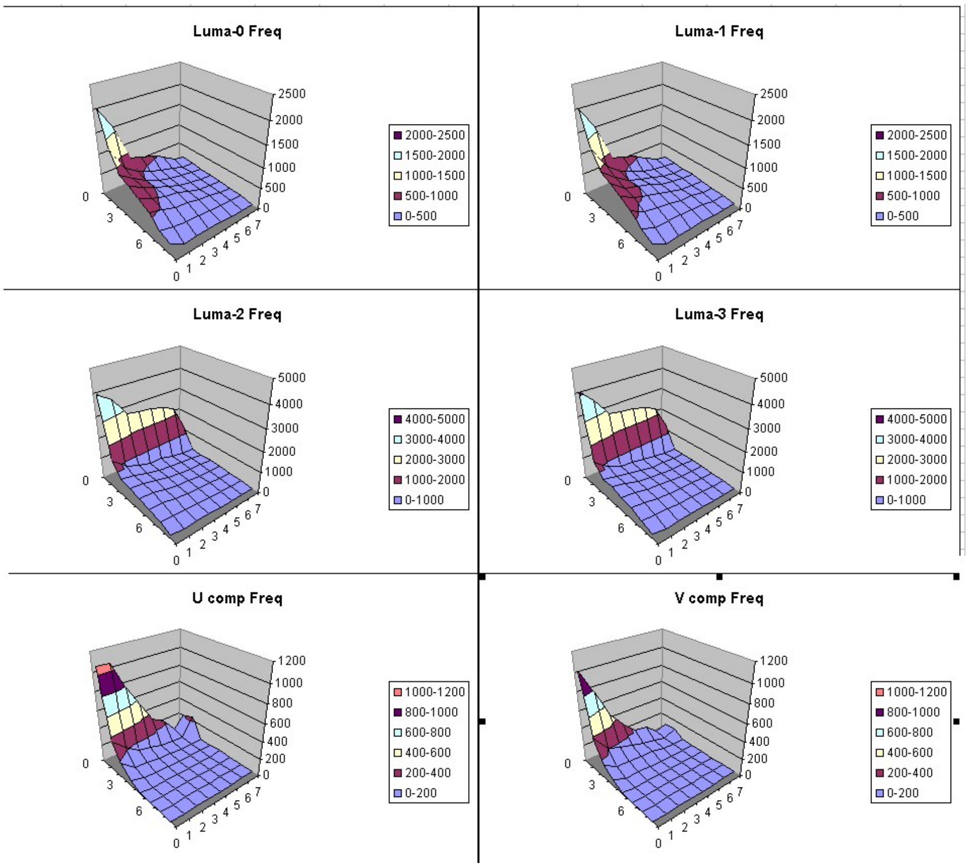
MV histogram. The motion vector histogram creates a graphical representation of the frequency of use of different motion vectors. This can be used to identify the range of search of the motion estimator in the encoder. The heights of the blocks represent the frequency with which the motion vectors were chosen.



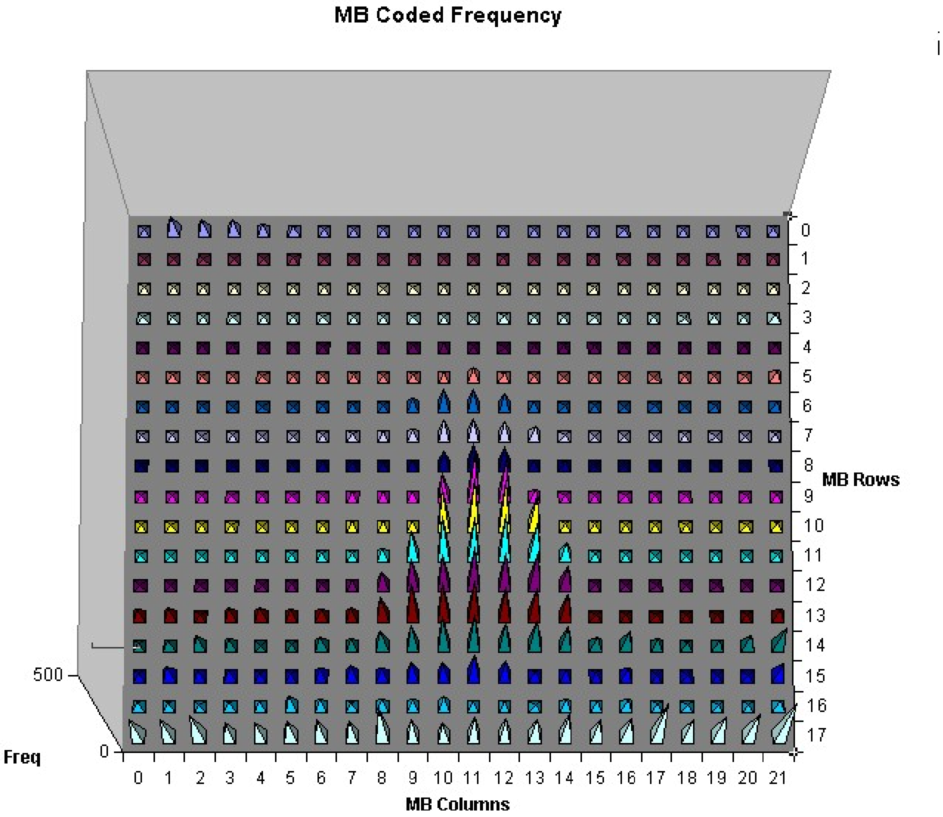
Spatial average quant. This option creates a graph of the quantizer choice at each part of the picture. This can be useful for determining poor quantizer control in certain encoders (often too low quantizers are chosen at the top of pictures, which then overshoots the bit-budget and then higher quantizers must be used progressively as the frame is encoded - meaning that the top of the picture is always coded more accurately than the bottom).



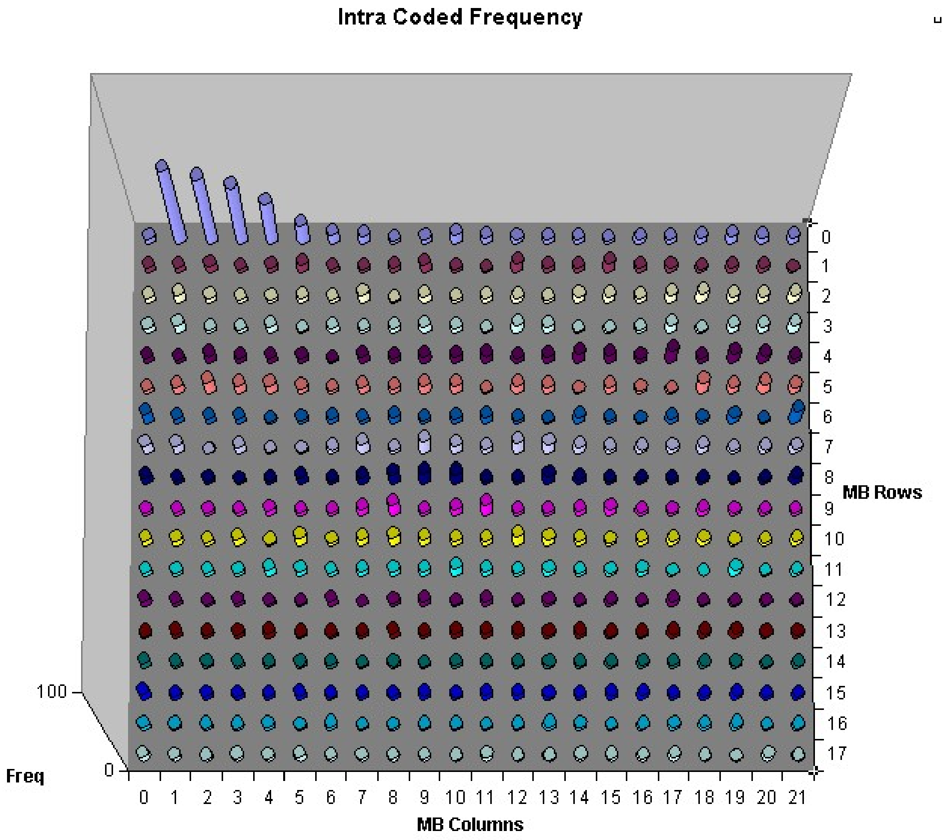
DCT frequency [MPEG 4, MPEG 2, and H.263 only]. This option creates a graph of the frequency with which the different components of the Discrete Transform Cosine are used. The positions of the transmitted coefficients are recorded and shown here for each of the six 8x8 blocks that make up the MacroBlock. This feature can be useful for identifying high frequency noise being transmitted in the picture as a result of poor front-end filtering.



MB coded frequency. This option creates a graph of the frequency of coding of each MacroBlock. It can help to identify the perpetual encoding of particular regions of the image due to issues in the digitization process, the rate control process, or the sensor.



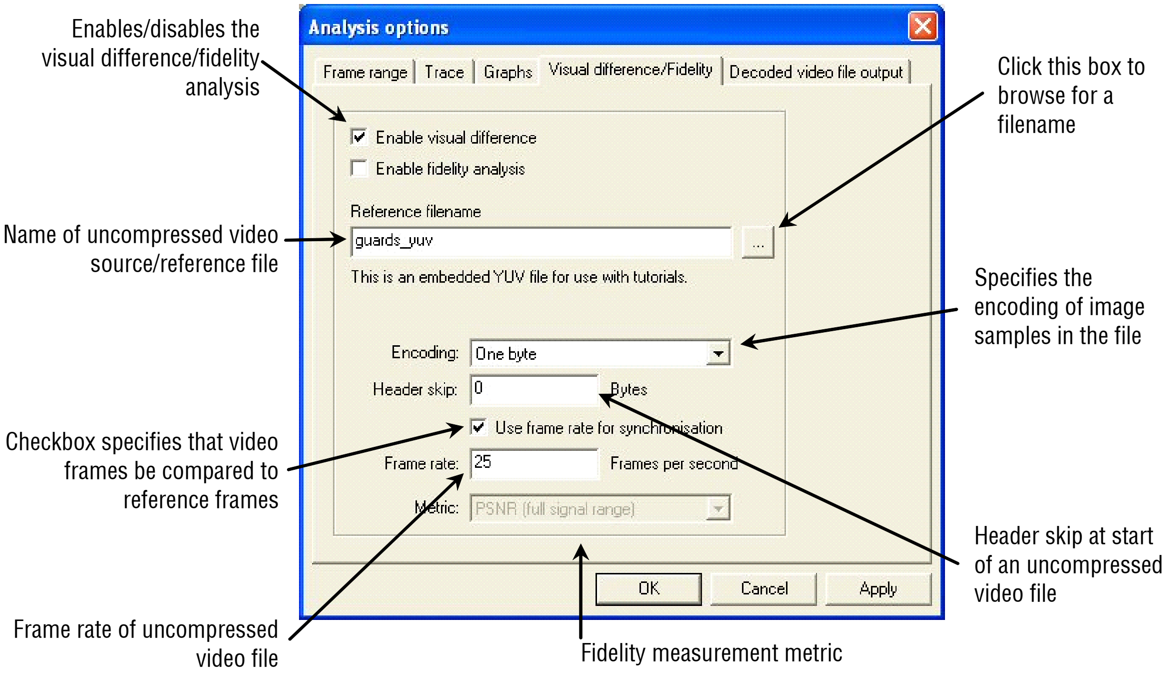
Intra coded frequency. This option creates a graph of the frequency of intra coding of each MacroBlock. Over time, each MacroBlock can be updated in Intra mode every so often (as required in the Standards) in case there is a line error or a mismatch between encoder and decoder. Encoders that do this more often are more likely to have better error robustness, although at the expense of more bits being used (as intra coding takes more bits than inter coding).



Video Fidelity Enable

NOTE. *Some options on this tab are different when H.264/AVC High Profile is used.*

This opens the Fidelity analysis tab of the Analysis options:



☑ Enable visual difference

□ Enable fidelity analysis

Reference filename: guards_yuv

This is an embedded YUV file for use with tutorials.

Encoding: One byte

Header skip: 0 Bytes

☑ Use frame rate for synchronisation

Frame rate: 25 Frames per second

Metric: PSNR (full signal range)

Enables/disables the visual difference/fidelity analysis

Name of uncompressed video source/reference file

Checkbox specifies that video frames be compared to reference frames

Frame rate of uncompressed video file

Click this box to browse for a filename

Specifies the encoding of image samples in the file

Header skip at start of an uncompressed video file

Fidelity measurement metric

The results of the fidelity analysis are displayed:

- In the Trace files, when the appropriate Trace option is selected (see *Video Trace Enable Ctrl+T* on page 6–136, *Frame fidelity* on page 6–152, and *MacroBlock fidelity* on page 6–152)
- As real-time overlays, when the overlay is selected (see *MB Statistics* on page 6–102, *Fidelity* on page 6–109, and *Average fidelity* on page 6–113)

Reference Filename

The name of the file used as the reference or source of encoding, of the encoded stream.

This file must be one of the following formats:

- 8 bits per sample, 4:2:0
- More than 8 bits per sample, and/or 4:2:2 or 4:4:4 (as used by H.264/AVC High Profile/FRExt, High/10, High/4:2:2, High/4:4:4)

YUV format of 8 bits per sample 4:2:0. The YUV file output is raw YUV with no headers of any kind: this is the same format as used by the Microsoft MPEG-4 Part 2 reference encoder Reference [7] in the *Compression Standards and File Types* section and as used commonly by other programs:

- No headers of any kind (no file or frame headers)
- One byte per sample
- Row raster order (top picture row first)
- Planar YUV 4:2:0 sub-sampled (4 bytes of Y data for each byte of U data and each byte of Y data)
- Y plane values are 0-255 unsigned
- U and V plane values are unsigned with a DC offset of 128

Other uncompressed formats. The general uncompressed video file format is as follows:

- No headers of any kind (no file or frame headers)
- Concatenated planar image data
- Row raster order (top picture row first)
- Unsigned samples

For 8-bit sample depth:

- One byte per sample

For 9-16 bit sample depth:

- Two bytes per sample
- Both little- and big-endian byte orders supported

For YUV format:

- Concatenated Y, U and V planes
- U and V planes sub-sampled as required
- Y plane samples are unsigned
- U and V plane samples are unsigned with a DC offset of 2^{n-1}, where *n* is the chroma sample bit depth

For RGB format:

- Concatenated R, G and B planes

For grayscale format:

- Luma plane only

NOTE. *There are built-in example YUV reference files for three of the example bitstreams: for MPEG-4 Man Walking, for H.264/AVC Grenadier Guards and for MPEG-2 Grenadier Guards. When these example streams are selected and fidelity analysis is enabled, the file names are automatically filled in. See* Example Files... *on page 6–37 for more information.*

Encoding

This control specifies the encoding of image samples in the file format. The following options are available:

- One byte - this specifies that image samples are stored in one byte per sample. This format is appropriate if all image planes are 8-bits deep
- Two byte MSB first - this specifies that image samples are stored in a pair of bytes for each sample. The most significant byte occurs first in each pair (big-endian). This format is appropriate if one or more image planes are deeper than 8-bits
- Two byte LSB first - this specifies that image samples are stored in a pair of bytes for each sample. The least significant byte occurs first in each pair (little-endian). This format is appropriate if one or more image planes are deeper than 8-bits

Header Skip

The number of bytes at the start of the file before the first frame: MTS4EA will skip past these bytes (ignoring them).

Use Frame Rate for Synchronization

This checkbox specifies that decoded video frames should be compared to reference frames according to corresponding time stamps. If this checkbox is not selected, then corresponding frame numbers are used.

Frame Rate

The rate at which the uncompressed video frames were recorded, in frames per second.

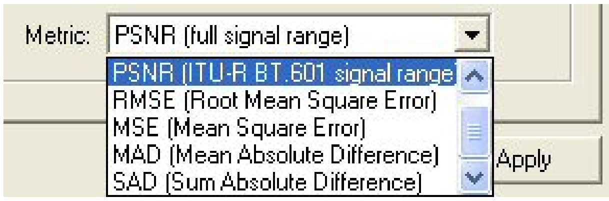
NOTE. *The number entered in the Frame rate field can be an integer (for example, 30), or a fraction (for example, 30000/1001) or a decimal number (for example, 29.97).*

Note that this frame rate is often different from the frame rate of the encoded video; for example, there can be frame skipping in the encoder, or a lower frame rate selected (such as encoding 30 frames/second video at 15 frames/second).

The Frame Rate value is used to determine which uncompressed video frame to associate with which encoded frame; MTS4EA works out the time code for each uncompressed video frame based upon the Frame Rate value and associates the uncompressed video frame concerned with the encoded frame which has the closest time code.

Metric

This selects the measurement metric to use, which is one of the following:



Metric: PSNR (full signal range)

PSNR (ITU-R BT.601 signal range)
RMSE (Root Mean Square Error)
MSE (Mean Square Error)
MAD (Mean Absolute Difference)
SAD (Sum Absolute Difference)
Apply

The objective fidelity metrics provided by MTS4EA measure the degradation of the decoded image with respect to a reference image. The metric is evaluated independently for each image plane in the color space dictated by the video decoder (typically YUV).

The fidelity metrics are calculated either per MacroBlock or overall on the frame as a whole, as per the following equations.

For the explanations of fidelity metrics, the following nomenclature is used:

| … |	Denotes taking the absolute value of an expression
Σ	Denotes the summation of an expression over the range of (x, y) in the image plane
$f(x, y)$	Is the sample value at the location (x, y) in the reference image plane
$g(x, y)$	Is the sample value at the location (x, y) in the decoded image plane
N	Is the total number of samples in the image plane

PSNR (full signal range). The PSNR (Peak Signal to Noise Ratio) metric is defined as the ratio between signal power and noise power, on a decibel scale. In the context of image processing, signal power is taken to be the square of the peak image sample value and noise power is taken as the square of RMS error in the image.

$$\text{PSNR} = 10 \cdot \log (\text{S2}/ \text{RMS2})$$

For PSNR (full signal range) the peak image value is assumed to be the maximum value for the bit-depth in all three image planes. For example, in an 8-bit image, this implies:

$$\text{PSNR255} = 20 \cdot \log (255/ \text{RMS})$$

PSNR (ITU-R BT.601 signal range). For PSNR (ITU-R BT.601 signal range) the PSNR is calculated differently in Y than in U and V:

$$\text{PSNR}_{\text{Y}} = 20 \cdot \log (220/ \text{RMS})$$

$$\text{PSNR}_{\text{U,V}} = 20 \cdot \log (225/ \text{RMS})$$

RMSE (Root Mean Square Error). The RMS is the defined as the square root of MSE.

$$\text{RMS} = \surd\, \text{MSE}$$

MSE (Mean Square Error). The MSE metric is defined as the mean of the squares of differences between samples in the reference and decoded image planes.

$$\text{MSE} = 1/N\, \Sigma\, [\, f(x, y) - g(x, y)\,]^2$$

MAD (Mean Absolute Difference). The MAD metric is defined as the mean average of absolute differences between samples in the reference and decoded image planes.

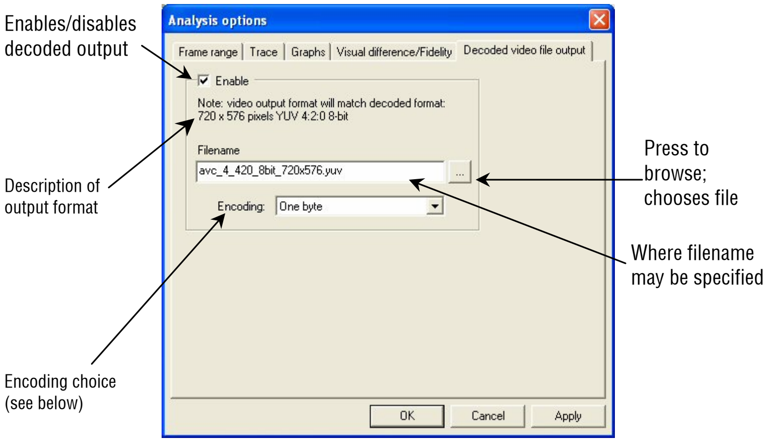
$$\text{MAD} = 1/N\, \Sigma\, |\, f(x, y) - g(x, y)\, |$$

SAD (Sum Absolute Difference). The SAD metric is defined as the sum of absolute differences between samples in the reference and decoded image planes.

$$\text{SAD} = \Sigma \mid f(x, y) - g(x, y) \mid$$

Output Decoded Video File...

This opens the Decoded output tab of the Analysis options and allows you to write a decoded file to disk.



☑ Enable

Note: video output format will match decoded format:
720 x 576 pixels YUV 4:2:0 8-bit

Filename: avc_4_420_8bit_720x576.yuv

Encoding: One byte

Enables/disables decoded output

Description of output format

Encoding choice (see below)

Press to browse; chooses file

Where filename may be specified

Encoding

This control specifies the encoding of image samples in the file format. The following options are available:

- One byte - this specifies that image samples are stored in one byte per sample. This format is appropriate if all image planes are 8-bits deep

- Two byte MSB first - this specifies that image samples are stored in a pair of bytes for each sample. The most significant byte occurs first in each pair (big-endian). This format is appropriate if one or more image planes are deeper than 8-bits

- Two byte LSB first - this specifies that image samples are stored in a pair of bytes for each sample. The least significant byte occurs first in each pair (little-endian). This format is appropriate if one or more image planes are deeper than 8-bits

Video Analysis Options...

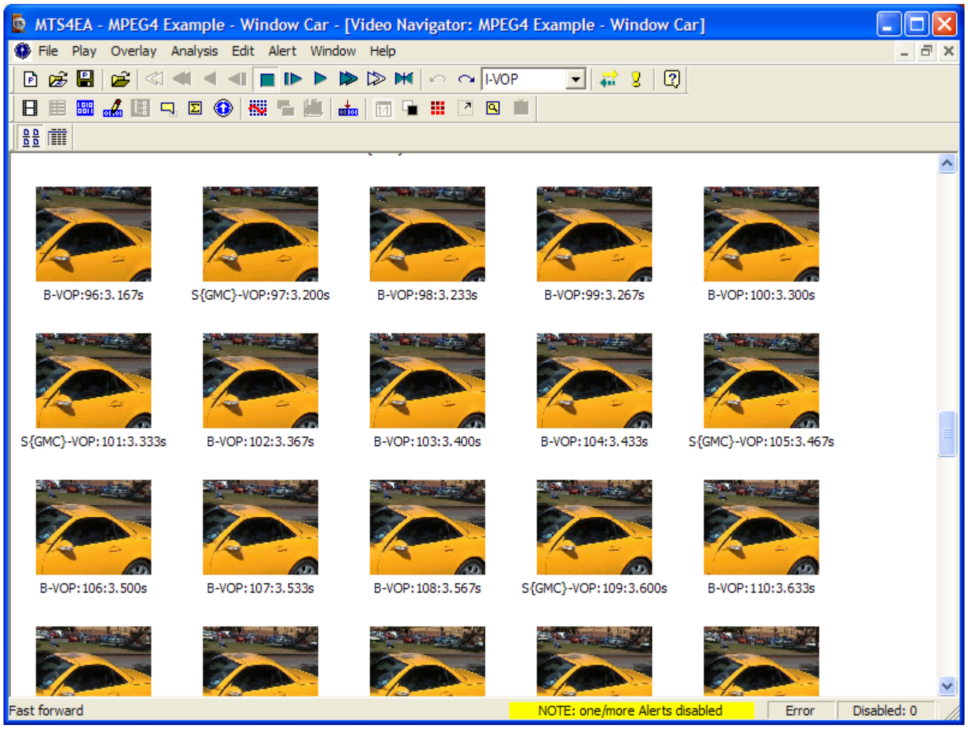
The Video Analysis menu option opens the Video Analysis dialog box; this allows you to select and set up the range of video analysis options.

View Video Navigator...

Thumbnail View

This shows a thumbnail view of the frames in a video file, with some basic information.

Two different views can be selected. In either case, a right-click context-sensitive menu allows selection of the other views and navigation to the corresponding frame in the other views (or the start address of the corresponding frame, as appropriate). See *Right-click popup menu/Goto views.* on page 6–170.



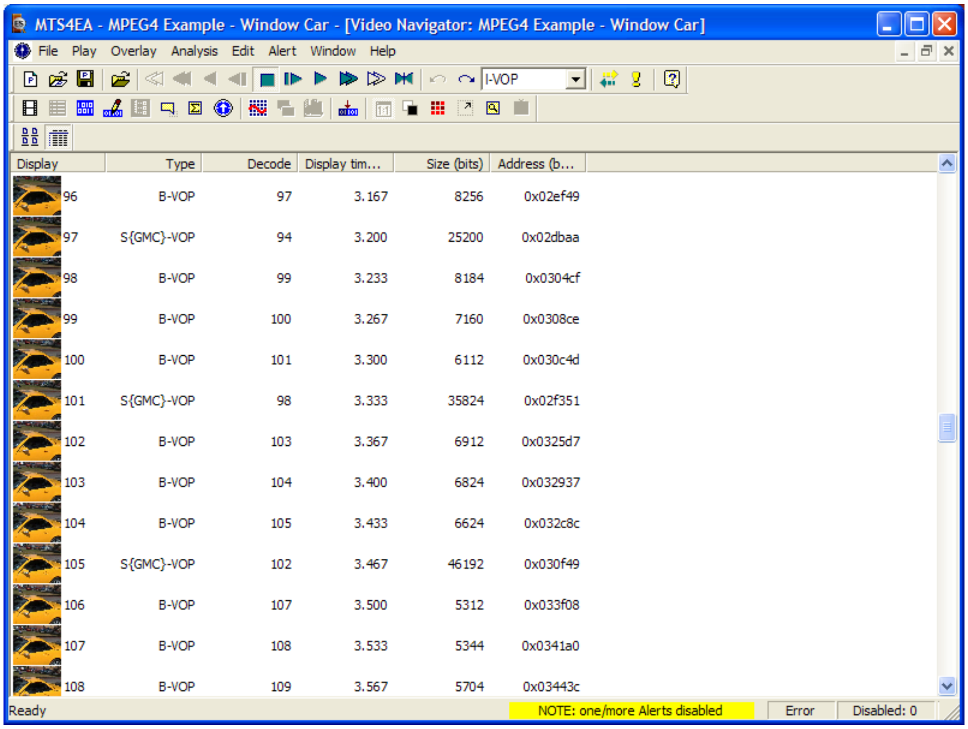
The video frames are shown in display order (decode order can be different if there are B-frames/B-VOPs).

The displayed information is:

- Frame type/VOP type
- Display frame/VOP number
- Display time in seconds

NOTE. *For interlaced bitstreams, the frame/VOP type reported is that relating to the second field in a frame. This means that if the top field is first in frame/VOP 1 (and this will always be Intra-coded) and the bottom field is Inter [P] coded, the frame will be reported as P.*

Detail View


MTS4EA - MPEG4 Example - Window Car - [Video Navigator: MPEG4 Example - Window Car]
File
Play
Overlay
Analysis
Edit
Alert
Window
Help
I-VOP
Ready
NOTE: one/more Alerts disabled
Error
Disabled: 0

Display	Type	Decode	Display tim...	Size (bits)	Address (b...
96	B-VOP	97	3.167	8256	0x02ef49
97	S{GMC}-VOP	94	3.200	25200	0x02dbaa
98	B-VOP	99	3.233	8184	0x0304cf
99	B-VOP	100	3.267	7160	0x0308ce
100	B-VOP	101	3.300	6112	0x030c4d
101	S{GMC}-VOP	98	3.333	35824	0x02f351
102	B-VOP	103	3.367	6912	0x0325d7
103	B-VOP	104	3.400	6824	0x032937
104	B-VOP	105	3.433	6624	0x032c8c
105	S{GMC}-VOP	102	3.467	46192	0x030f49
106	B-VOP	107	3.500	5312	0x033f08
107	B-VOP	108	3.533	5344	0x0341a0
108	B-VOP	109	3.567	5704	0x03443c

The displayed information is:

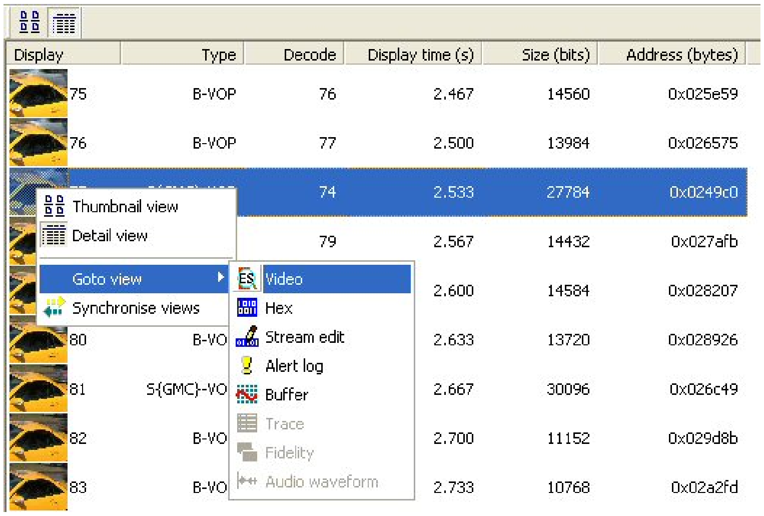
- Display frame/VOP number
- Frame type/VOP type

NOTE. *For interlaced bitstreams, the reported frame/VOP type is that relating to the second field in a frame. This means that if the top field is first in frame/VOP 1 (and this will always be Intra-coded) and the bottom field is Inter [P] coded, the frame will be reported as P.*

- Decode frame/VOP number
- Display time in seconds
- Frame size in bits
- Address of start of frame (in the video elementary stream, not in the overall container file if there is one)

The columns may be resized and dragged into a different order if desired.

Right-click popup menu/Goto views.



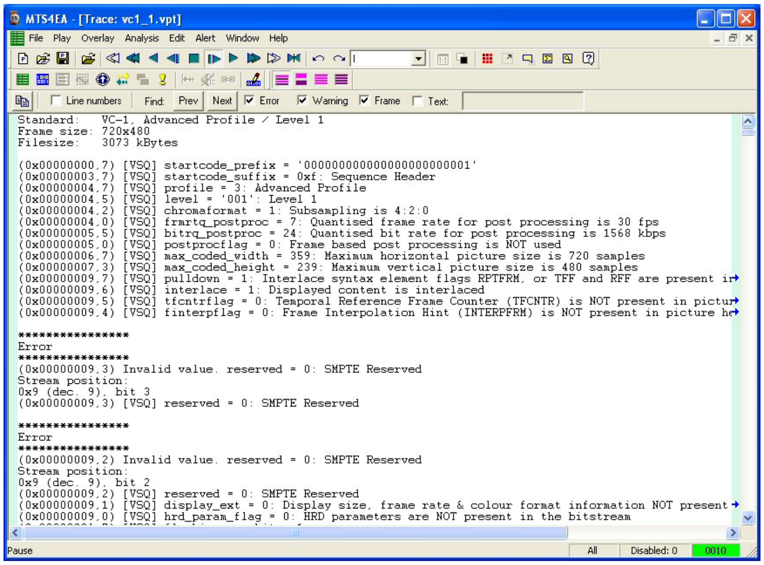
Display	Type	Decode	Display time (s)	Size (bits)	Address (bytes)
75	B-VOP	76	2.467	14560	0x025e59
76	B-VOP	77	2.500	13984	0x026575
		74	2.533	27784	0x0249c0
		79	2.567	14432	0x027afb
			2.600	14584	0x028207
80	B-VO		2.633	13720	0x028926
81	S{GMC}-VO		2.667	30096	0x026c49
82	B-VO		2.700	11152	0x029d8b
83	B-VO		2.733	10768	0x02a2fd

See *Synchronized Views/Navigating the Views* on page 6–7 for more information on navigating between views.

View Video Trace... ▤ Ctrl+Shift+T

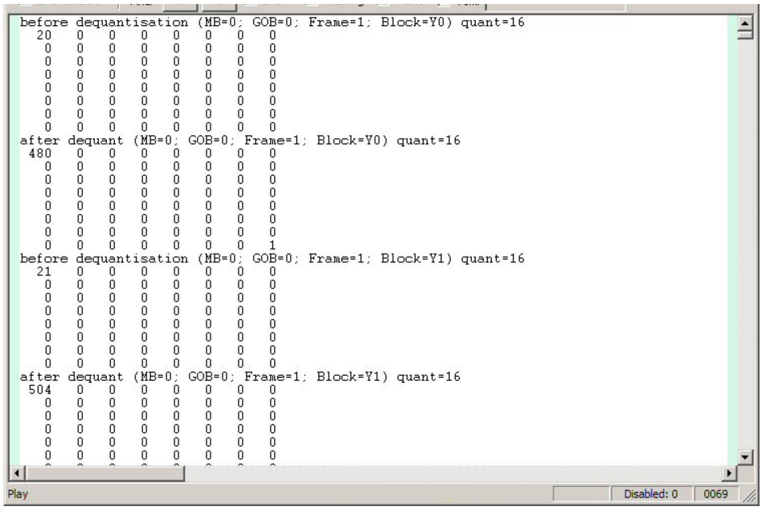
This option opens the Trace view on the current output Trace file, as shown in the Analysis menu Trace tab.

The screenshot below shows a Trace/Interpret file, but all of the Trace outputs are viewed in the Trace view:



```
Standard:   VC-1, Advanced Profile / Level 1
Frame size: 720x480
Filesize:   3073 kBytes

(0x00000000,7) [VSQ] startcode_prefix = '000000000000000000000001'
(0x00000003,7) [VSQ] startcode_suffix = 0xf: Sequence Header
(0x00000004,7) [VSQ] profile = 3: Advanced Profile
(0x00000004,5) [VSQ] level = '001': Level 1
(0x00000004,2) [VSQ] chromaformat = 1: Subsampling is 4:2:0
(0x00000004,0) [VSQ] frmrtq_postproc = 7: Quantised frame rate for post processing is 30 fps
(0x00000005,5) [VSQ] bitrq_postproc = 24: Quantised bit rate for post processing is 1568 kbps
(0x00000005,0) [VSQ] postprocflag = 0: Frame based post processing is NOT used
(0x00000006,7) [VSQ] max_coded_width = 359: Maximum horizontal picture size is 720 samples
(0x00000007,3) [VSQ] max_coded_height = 239: Maximum vertical picture size is 480 samples
(0x00000009,7) [VSQ] pulldown = 1: Interlace syntax element flags RPTFRM, or TFF and RFF are present ir→
(0x00000009,6) [VSQ] interlace = 1: Displayed content is interlaced
(0x00000009,5) [VSQ] tfcntrflag = 0: Temporal Reference Frame Counter (TFCNTR) is NOT present in pictur→
(0x00000009,4) [VSQ] finterpflag = 0: Frame Interpolation Hint (INTERPFRM) is NOT present in picture he→

****************
Error
****************
(0x00000009,3) Invalid value. reserved = 0: SMPTE Reserved
Stream position:
0x9 (dec. 9), bit 3
(0x00000009,3) [VSQ] reserved = 0: SMPTE Reserved

****************
Error
****************
(0x00000009,2) Invalid value. reserved = 0: SMPTE Reserved
Stream position:
0x9 (dec. 9), bit 2
(0x00000009,2) [VSQ] reserved = 0: SMPTE Reserved
(0x00000009,1) [VSQ] display_ext = 0: Display size, frame rate & colour format information NOT present →
(0x00000009,0) [VSQ] hrd_param_flag = 0: HRD parameters are NOT present in the bitstream
```

This is a Trace/DCT level example:



```
before dequantisation (MB=0; GOB=0; Frame=1; Block=Y0) quant=16
  20    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
after dequant (MB=0; GOB=0; Frame=1; Block=Y0) quant=16
 480    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    1
before dequantisation (MB=0; GOB=0; Frame=1; Block=Y1) quant=16
  21    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
after dequant (MB=0; GOB=0; Frame=1; Block=Y1) quant=16
 504    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0
```

NOTE. *If there is not a current Trace file (and Trace enable is not enabled) then this menu option/toolbar icon is unavailable (grayed out).*

NOTE. *For users of earlier versions of the MTS4EA, this Trace viewer replaces the previous Trace view done using Microsoft Word.*

If the video stream has been rerun and a Trace file saved with the same name as that already open, then clicking the View trace... menu option or ▦ icon or `Ctrl+V` will refresh the open Trace file.

Go to other views/right click menu - Interpret & Parse Bitstream only

When you right-click, a context-sensitive menu appears that allows going to the area in another view which corresponds to the area which has been selected in the Trace file:

Copying Text

The contents of the Trace window can be copied to the Windows clipboard.

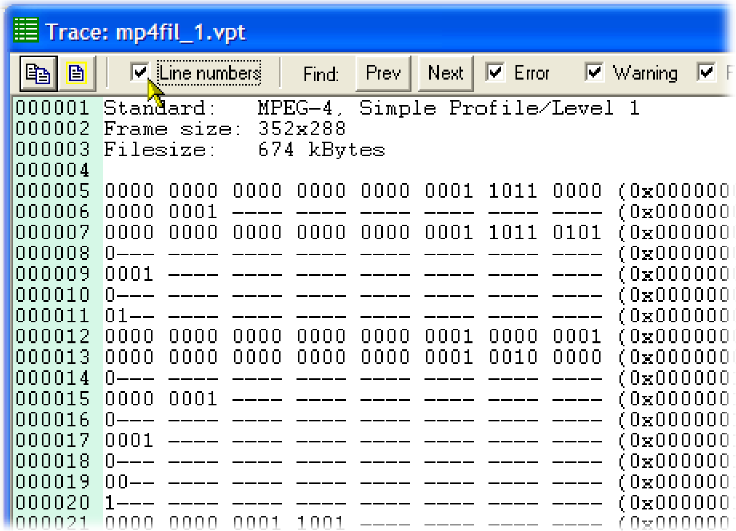
This can be done in a number of ways:

- By pressing `Ctrl+C`
- By right-clicking with the mouse and selecting Copy
- By clicking the copy icon - 📋

The whole of the contents can be selected by clicking the Select All icon - 🗎.

Line Numbers

Line numbers can be turned on/off:



```
000001 Standard:   MPEG-4, Simple Profile/Level 1
000002 Frame size: 352x288
000003 Filesize:   674 kBytes
000004
000005 0000 0000 0000 0000 0000 0001 1011 0000 (0x000000
000006 0000 0001 ---- ---- ---- ---- ---- ---- (0x000000
000007 0000 0000 0000 0000 0000 0001 1011 0101 (0x000000
000008 0--- ---- ---- ---- ---- ---- ---- ---- (0x000000
000009 0001 ---- ---- ---- ---- ---- ---- ---- (0x000000
000010 0--- ---- ---- ---- ---- ---- ---- ---- (0x000000
000011 01-- ---- ---- ---- ---- ---- ---- ---- (0x000000
000012 0000 0000 0000 0000 0000 0001 0000 0001 (0x000000
000013 0000 0000 0000 0000 0000 0001 0010 0000 (0x000000
000014 0--- ---- ---- ---- ---- ---- ---- ---- (0x000000
000015 0000 0001 ---- ---- ---- ---- ---- ---- (0x000000
000016 0--- ---- ---- ---- ---- ---- ---- ---- (0x000000
000017 0001 ---- ---- ---- ---- ---- ---- ---- (0x000000
000018 0--- ---- ---- ---- ---- ---- ---- ---- (0x000000
000019 00-- ---- ---- ---- ---- ---- ---- ---- (0x000000
000020 1--- ---- ---- ---- ---- ---- ---- ---- (0x000000
000021 0000 0000 0001 1001 ---- ---- ---- ---- (0x000000
```

Find Data

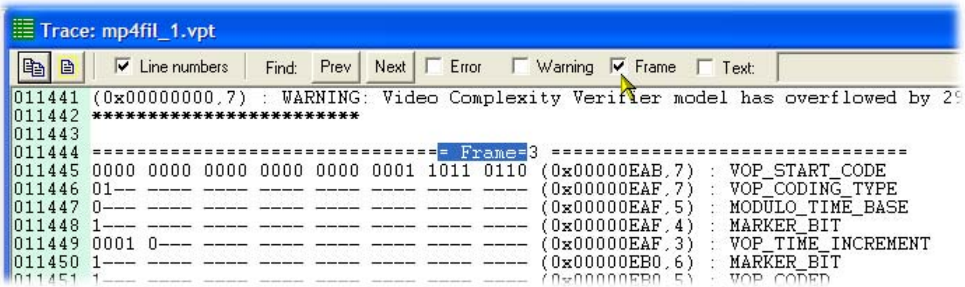
The Trace file can be searched for any data, using the Find: Prev (previous) and Next buttons. This finds the previous/next occurrence of any of the enabled strings.

NOTE. *The F3 key can also be pressed, to find next. The Shift+F3 key can also be pressed, to find previous.*

There are some standard strings that are useful to search for:

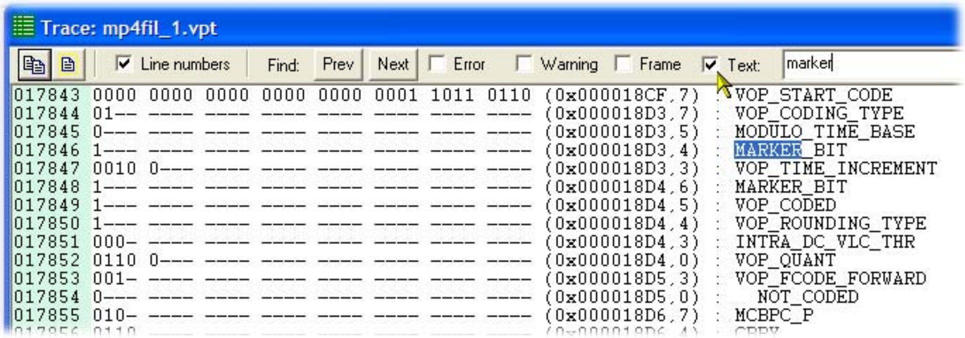
- Error
- Warning
- Frame

These are found by enabling the relevant check box (Frame example below):



```
011441 (0x00000000,7) : WARNING: Video Complexity Verifier model has overflowed by 29
011442 ************************
011443
011444 ==============================  Frame=3 ===============================
011445 0000 0000 0000 0000 0000 0001 1011 0110 (0x00000EAB,7) : VOP_START_CODE
011446 01-- ---- ---- ---- ---- ---- ---- ---- (0x00000EAF,7) : VOP_CODING_TYPE
011447 0--- ---- ---- ---- ---- ---- ---- ---- (0x00000EAF,5) : MODULO_TIME_BASE
011448 1--- ---- ---- ---- ---- ---- ---- ---- (0x00000EAF,4) : MARKER_BIT
011449 0001 0--- ---- ---- ---- ---- ---- ---- (0x00000EAF,3) : VOP_TIME_INCREMENT
011450 1--- ---- ---- ---- ---- ---- ---- ---- (0x00000EB0,6) : MARKER_BIT
011451 1--- ---- ---- ---- ---- ---- ---- ---- (0x00000EB0,5) : VOP_CODED
```

To search for any text, select the box next to Text and enter the text in the box (in this example, searching for `marker`):



```
017843 0000 0000 0000 0000 0000 0001 1011 0110 (0x000018CF,7) : VOP_START_CODE
017844 01-- ---- ---- ---- ---- ---- ---- ---- (0x000018D3,7) : VOP_CODING_TYPE
017845 0--- ---- ---- ---- ---- ---- ---- ---- (0x000018D3,5) : MODULO_TIME_BASE
017846 1--- ---- ---- ---- ---- ---- ---- ---- (0x000018D3,4) : MARKER_BIT
017847 0010 0--- ---- ---- ---- ---- ---- ---- (0x000018D3,3) : VOP_TIME_INCREMENT
017848 1--- ---- ---- ---- ---- ---- ---- ---- (0x000018D4,6) : MARKER_BIT
017849 1--- ---- ---- ---- ---- ---- ---- ---- (0x000018D4,5) : VOP_CODED
017850 1--- ---- ---- ---- ---- ---- ---- ---- (0x000018D4,4) : VOP_ROUNDING_TYPE
017851 000- ---- ---- ---- ---- ---- ---- ---- (0x000018D4,3) : INTRA_DC_VLC_THR
017852 0110 0--- ---- ---- ---- ---- ---- ---- (0x000018D4,0) : VOP_QUANT
017853 001- ---- ---- ---- ---- ---- ---- ---- (0x000018D5,3) : VOP_FCODE_FORWARD
017854 0--- ---- ---- ---- ---- ---- ---- ---- (0x000018D5,0) :   NOT_CODED
017855 010- ---- ---- ---- ---- ---- ---- ---- (0x000018D6,7) : MCBPC_P
017856 0110 ---- ---- ---- ---- ---- ---- ---- (0x000018D6,4) : CBPY
```

Lines too long for Window Width

If a line is too long to be displayed within the width of the View trace window, then a horizontal scroll bar is displayed and a blue arrow is appended to the right-hand side of the View trace window, to indicate that a line extends to the right.

Trace File Format

The MTS4EA saves Trace files with a `.vpt` file extension.

Currently, the .vpt file is a standard ASCII file, which could be viewed in any text file viewer. The .vpt extension is used:

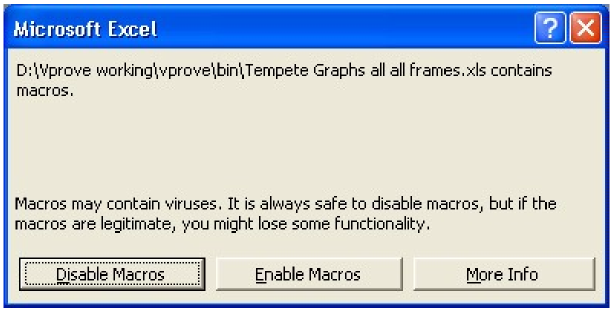
- So that this file extension is associated with the MTS4EA in Windows, and by double-clicking on a .vpt file in Windows Explorer, the file is automatically opened in the MTS4EA

View Video Graphs...

This option calls Microsoft Excel to display the graphs generated: a macro is then run within Excel to create the graphs.

NOTE. *Macros must be enabled within Microsoft Excel to see the graphs.*

The default selection within Microsoft Excel is (usually) to disable macros (as below) - you must click on Enable Macros:



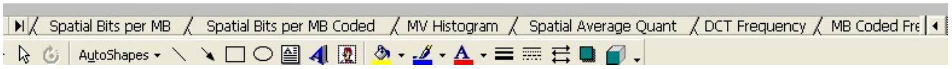
Microsoft Excel

D:\Vprove working\vprove\bin\Tempete Graphs all all frames.xls contains macros.

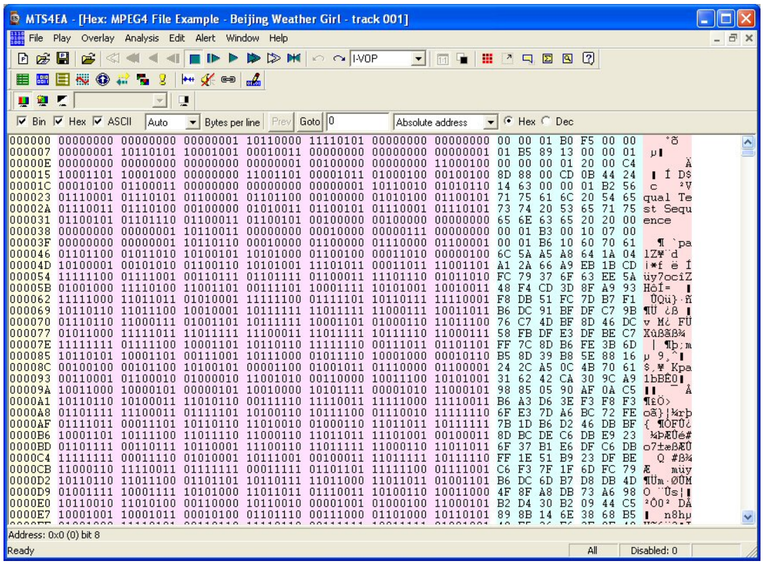
Macros may contain viruses. It is always safe to disable macros, but if the macros are legitimate, you might lose some functionality.

NOTE. *The data used to generate the graphs is available on the Data tab in the Excel file.*

The statistics options selected on the Graph enable tab within MTS4EA are then displayed on individual tabs within Excel:



View Video Stream Hex... Ctrl+H



MTS4EA - [Hex: MPEG4 File Example - Beijing Weather Girl - track 001]

☑ Bin ☑ Hex ☑ ASCII Auto Bytes per line Prev Goto 0 Absolute address ● Hex ○ Dec

```
000000 00000000 00000000 00000001 10110000 11110101 00000000 00000000 00 00 01 B0 F5 00 00    °õ
000007 00000001 10110101 10001001 00010011 00000000 00000000 00000001 01 B5 89 13 00 00 01  µ▮
00000E 00000000 00000000 00000000 00000001 00100000 00000000 11000100 00 00 00 01 20 00 C4       Ä
000015 10001101 10001000 00000000 11001101 00001011 01000100 00100100 8D 88 00 CD 0B 44 24  ▮ Í D$
00001C 00010100 01100011 00000000 00000000 00000001 10110010 01010110 14 63 00 00 01 B2 56  c   ²V
000023 01110001 01110101 01100001 01101100 00100000 01010100 01100101 71 75 61 6C 20 54 65 qual Te
00002A 01110011 01110100 00100000 01010011 01100101 01110001 01110101 73 74 20 53 65 71 75 st Sequ
000031 01100101 01101110 01100011 01100101 00100000 00100000 00000000 65 6E 63 65 20 20 00 ence
000038 00000000 00000001 10110011 00000000 00010000 00000111 00000000 00 01 B3 00 10 07 00
00003F 00000000 00000001 10110110 00010000 01100000 01110000 01100001 00 01 B6 10 60 70 61   ¶ `pa
000046 01101100 01011010 10100101 10101000 01100100 00011010 00000100 6C 5A A5 A8 64 1A 04 lZ¥¨d
00004D 10100001 00101010 01100110 10101001 11101011 00011011 11001101 A1 2A 66 A9 EB 1B CD ¡*f ë Í
000054 11111100 01111001 00110111 01101111 01100011 11101110 01011010 FC 79 37 6F 63 EE 5A üy7ocîZ
00005B 01001000 11110100 11001101 00111101 10001111 10101001 10010011 48 F4 CD 3D 8F A9 93 HôÍ=  ▮
000062 11111000 11011011 01010001 11111100 01111101 10110111 11110001 F8 DB 51 FC 7D B7 F1  ÛQü}·ñ
000069 10110110 11011100 10010001 10111111 11011111 11000111 10011011 B6 DC 91 BF DF C7 9B ¶Ü ¿ß ▮
000070 01110110 11000111 01001101 10111111 10001101 01000110 11011100 76 C7 4D BF 8D 46 DC v M¿ FÜ
000077 01011000 11111011 11011111 11100011 11011111 10111110 11000111 58 FB DF E3 DF BE C7 Xûßãß¾
00007E 11111111 01111100 10001101 10110110 11111110 00111011 01101101 FF 7C 8D B6 FE 3B 6D  | ¶þ;m
000085 10110101 10001101 00111001 10111000 01011110 10001000 00010110 B5 8D 39 B8 5E 88 16 µ 9¸^▮
00008C 00100100 00101100 10100101 00001100 01001011 01110000 01100001 24 2C A5 0C 4B 70 61 $,¥ Kpa
000093 00110001 01100010 01000010 11001010 00110000 10011100 10101001 31 62 42 CA 30 9C A9 1bBÊ0▮
00009A 10011000 10000101 00000101 10010000 10101111 00001010 11000101 98 85 05 90 AF 0A C5 ▮▮  ¯ Å
0000A1 10110110 10100011 11010110 00111110 11110011 11111000 11110011 B6 A3 D6 3E F3 F8 F3 ¶£Ö>
0000A8 01101111 11100011 01111101 10100110 10111100 01110010 11111110 6F E3 7D A6 BC 72 FE oã}¦¼rþ
0000AF 01111011 00011101 10110110 11010010 01000110 11011011 10111111 7B 1D B6 D2 46 DB BF { ¶ÒFÛ¿
0000B6 10001101 10111100 11011110 11000110 11011011 11101001 00100011 8D BC DE C6 DB E9 23  ¼ÞÆÛé#
0000BD 01101111 00110111 10110001 11100110 11011111 11000110 11011011 6F 37 B1 E6 DF C6 DB o7±æßÆÛ
0000C4 11111111 00011110 01010001 10111001 00100011 11011111 10111110 FF 1E 51 B9 23 DF BE   Q #ß¾
0000CB 11000110 11110011 01111111 00011111 01101101 11111100 01111001 C6 F3 7F 1F 6D FC 79 Æ   müy
0000D2 10110110 11011100 01101101 10110111 11011000 11011011 01001101 B6 DC 6D B7 D8 DB 4D ¶Üm·ØÛM
0000D9 01001111 10001111 10101000 11011011 01110011 10100110 10011000 4F 8F A8 DB 73 A6 98 O ¨Ûs¦▮
0000E0 10110010 11010100 00110000 10110010 00001001 01000100 11000101 B2 D4 30 B2 09 44 C5 ²Ô0² DÅ
0000E7 10001001 10001011 00010100 01101110 00111000 01101000 10110101 89 8B 14 6E 38 68 B5 ▮  n8hµ
```

Address: 0x0 (0) bit 8

Ready All Disabled: 0

This opens the current video file in a hex viewer, which shows the data in binary, hexadecimal and ASCII data (in any combination of the three).

NOTE. *Multiple View hex... windows can be opened at the same time.*

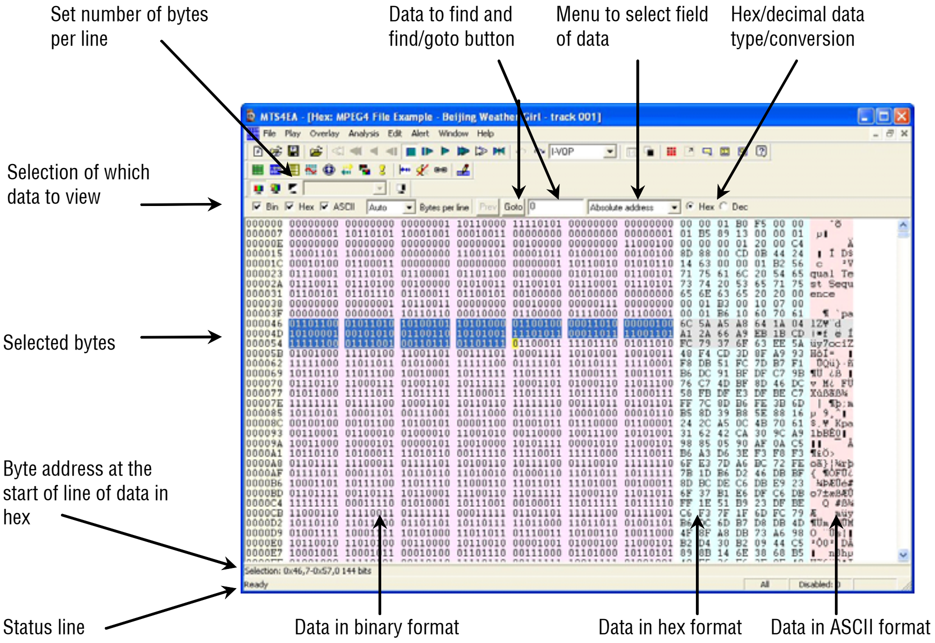
Using HexView, you can scroll through the file, and search for specific:

- Absolute address (from the start of the file)
- Relative address (from the currently selected location
- Bit patterns
- Hex data
- ASCII data

NOTE. *In the field where the data to find is entered, a wildcard character can be entered - this is . [period]*

Also, entering a hex value, then clicking Dec will convert this number to decimal (and vice versa).

Sections of HexView Window


Set number of bytes per line
Data to find and find/goto button
Menu to select field of data
Hex/decimal data type/conversion
Selection of which data to view
Selected bytes
Byte address at the start of line of data in hex
Status line
Data in binary format
Data in hex format
Data in ASCII format
MTS4EA - [Hex: MPEG4 File Example - Beijing Weather Girl - track 001]
File
Play
Overlay
Analysis
Edit
Alert
Window
Help
I-VOP
☑ Bin
☑ Hex
☑ ASCII
Auto
Bytes per line
Prev
Goto
0
Absolute address
● Hex
○ Dec
All
Disabled: 0

```
000000 00000000 00000000 00000001 10110000 11110101 00000000 00000000 00 00 01 B0 F5 00 00
000007 00000001 10110101 10001001 00010011 00000000 00000000 00000001 01 B5 89 13 00 00 01
00000E 00000000 00000000 00000000 00000001 00100000 00000000 11000100 00 00 00 01 20 00 C4
000015 10001101 10001000 00000000 11001101 00001011 01000100 00100100 8D 88 00 CD 0B 44 24
00001C 00010100 01100011 00000000 00000000 00000001 10110010 01010110 14 63 00 00 01 B2 56
000023 01110001 01110101 01100001 01101100 00100000 01010100 01100101 71 75 61 6C 20 54 65 qual Te
00002A 01110011 01110100 00100000 01010011 01100101 01110001 01110101 73 74 20 53 65 71 75 st Sequ
000031 01100101 01101110 01100011 01100101 00100000 00100000 00000000 65 6E 63 65 20 20 00 ence
000038 00000000 00000001 10110011 00000000 00010000 00000111 00000000 00 01 B3 00 10 07 00
00003F 00000000 00000001 10110110 00010000 01100000 01110000 01100001 00 01 B6 10 60 70 61   ¶ `pa
000046 01101100 01011010 10100101 10101000 01100100 00011010 00000100 6C 5A A5 A8 64 1A 04 lZ¥¨d
00004D 10100001 00101010 01100110 10101001 11101011 00011011 11001101 A1 2A 66 A9 EB 1B CD
000054 11111100 01111001 00110111 01101111 01100011 11101110 01011010 FC 79 37 6F 63 EE 5A üy7ocîZ
00005B 01001000 11110100 11001101 00111101 10001111 10101001 10010011 48 F4 CD 3D 8F A9 93
000062 11111000 11011011 01010001 11111100 01111101 10110111 11110001 F8 DB 51 FC 7D B7 F1
000069 10110110 11011100 10010001 10111111 11011111 11000111 10011011 B6 DC 91 BF DF C7 9B
000070 01110110 11000111 01001101 10111111 10001101 01000110 11011100 76 C7 4D BF 8D 46 DC
000077 01011000 11111011 11011111 11100011 11011111 10111110 11000111 58 FB DF E3 DF BE C7
00007E 11111111 01111100 10001101 10110110 11111110 00111011 01101101 FF 7C 8D B6 FE 3B 6D
000085 10110101 10001101 00111001 10111000 01011110 10001000 00010110 B5 8D 39 B8 5E 88 16
00008C 00100100 00101100 10100101 00001100 01001011 01110000 01100001 24 2C A5 0C 4B 70 61
000093 00110001 01100010 01000010 11001010 00110000 10011100 10101001 31 62 42 CA 30 9C A9
00009A 10011000 10000101 00000101 10010000 10101111 00001010 11000101 98 85 05 90 AF 0A C5
0000A1 10110110 10100011 11010110 00111110 11110011 11111000 11110011 B6 A3 D6 3E F3 F8 F3
0000A8 01101111 11100011 01111101 10100110 10111100 01110010 11111110 6F E3 7D A6 BC 72 FE
0000AF 01111011 00011101 10110110 11010010 01000110 11011011 10111111 7B 1D B6 D2 46 DB BF
0000B6 10001101 10111100 11011110 11000110 11011011 11101001 00100011 8D BC DE C6 DB E9 23
0000BD 01101111 00110111 10110001 11100110 11011111 11000110 11011011 6F 37 B1 E6 DF C6 DB
0000C4 11111111 00011110 01010001 10111001 00100011 11011111 10111110 FF 1E 51 B9 23 DF BE
0000CB 11000110 11110011 01111111 00011111 01101101 11111100 01111001 C6 F3 7F 1F 6D FC 79
0000D2 10110110 11011100 01101101 10110111 11011000 11011011 01001101 B6 DC 6D B7 D8 DB 4D
0000D9 01001111 10001111 10101000 11011011 01110011 10100110 10011000 4F 8F A8 DB 73 A6 98
0000E0 10110010 11010100 00110000 10110010 00001001 01000100 11000101 B2 D4 30 B2 09 44 C5
0000E7 10001001 10001011 00010100 01101110 00111000 01101000 10110101 89 8B 14 6E 38 68 B5
```

Selection: 0x46,7-0x57,0 144 bits

Ready

Setting Information Displayed; Window Width

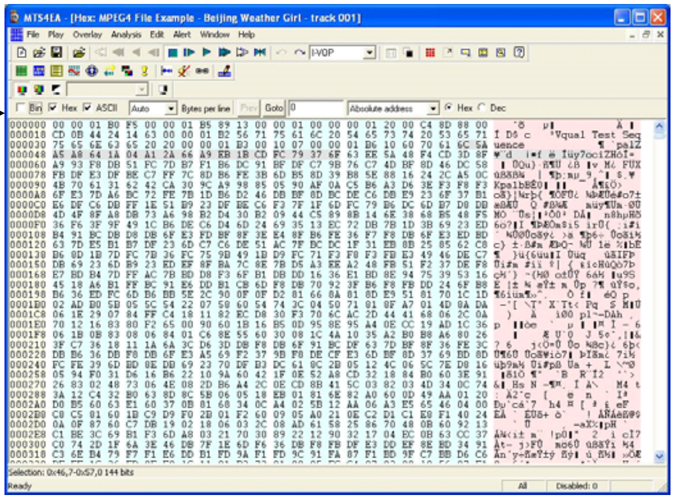
The HexView window can be set (for example, resized) as with any other window in Windows.

The same data is shown in each of the three sections, in their respective formats:

- Binary
- Hexadecimal
- ASCII

Each of these areas can be individually displayed/not displayed by clicking the appropriate check box in the top left.

In this example, the binary display has been switched off, leaving only hex and ASCII


MTS4EA - [Hex: MPEG4 File Example - Beijing Weather Girl - track 001]
File Play Overlay Analysis Edit Alert Window Help
I-VOP
□ Bin ☑ Hex ☑ ASCII
Auto
Bytes per line
Prev
Goto
0
Absolute address
● Hex ○ Dec
All
Disabled: 0

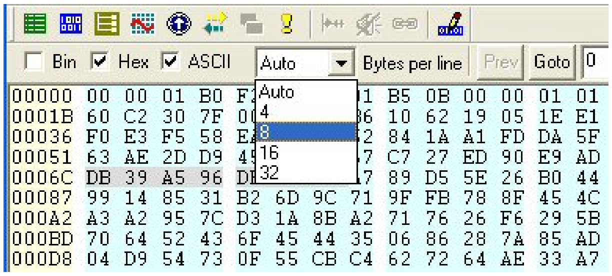
```
000000 00 00 01 B0 F5 00 00 01 B5 89 13 00 00 01 00 00 00 01 20 00 C4 8D 88 00
000018 CD 0B 44 24 14 63 00 00 01 B2 56 71 75 61 6C 20 54 65 73 74 20 53 65 71
000030 75 65 6E 63 65 20 20 00 00 01 B3 00 10 07 00 00 01 B6 10 60 70 61 6C 5A
000048 A5 A8 64 1A 04 A1 2A 66 A9 EB 1B CD FC 79 37 6F 63 EE 5A 48 F4 CD 3D 8F
000060 A9 93 F8 DB 51 FC 7D B7 F1 B6 DC 91 BF DF C7 9B 76 C7 4D BF 8D 46 DC 58
000078 FB DF E3 DF BE C7 FF 7C 8D B6 FE 3B 6D B5 8D 39 B8 5E 88 16 24 2C A5 0C
000090 4B 70 61 31 62 42 CA 30 9C A9 98 85 05 90 AF 0A C5 B6 A3 D6 3E F3 F8 F3
0000A8 6F E3 7D A6 BC 72 FE 7B 1D B6 D2 46 DB BF 8D BC DE C6 DB E9 23 6F 37 B1
0000C0 E6 DF C6 DB FF 1E 51 B9 23 DF DE C6 F3 7F 1F 6D FC 79 B6 DC 6D B7 D8 DB
0000D8 4D 4F 8F A8 DB 73 A6 98 B2 D4 30 B2 09 44 C5 89 8B 14 6E 38 68 B5 48 F5
0000F0 36 F6 3F 9F 49 1C B6 DE C6 D4 6D 24 69 35 13 EC 72 DB 7B 1D 3B 69 23 ED
000108 B4 91 BC DB D8 DB 6F E3 FD BF 8F 3E E4 8F B6 FE 36 F7 F8 DB 6F E3 ED BD
000120 63 7D E5 B1 B7 DF 23 6D C7 C6 DE 51 AC 7F BC DC 1F 31 EB 0B 25 05 62 C8
000138 B6 0D 1B 7D FC 7B 36 FC 75 9D 49 1B D9 FC 71 F3 F0 F3 FB E3 49 46 DE C7
000150 DB 69 23 6D B9 23 ED EF 8F BA 7C 8E 7B D5 A3 EE A2 48 FB 51 F2 37 DE F8
000168 E7 BD B4 7D FF AC 7B BD D8 F3 6F B1 DB DD 16 36 E1 BD 8E 94 75 39 53 16
000180 45 18 A6 B1 FF BC 91 E6 DD B1 CB 6D F8 DB 70 92 3F B6 F8 FB DD 24 6F B8
000198 B6 36 ED FC 6D D6 BB 5E 2C 90 0F 0F D2 01 66 0A 01 8D E9 51 81 70 1C 1D
0001B0 02 AD D0 5B 05 5C 54 22 07 58 60 54 74 3C 04 50 71 01 0F A7 01 4D 0A DA
0001C8 06 1E 29 07 84 FF C4 18 11 82 EC D8 30 F3 70 6C AC 2D 44 41 68 06 2C 0A
0001E0 70 12 16 83 80 F2 65 00 90 60 1B 16 B5 0D 95 8E 95 A4 0E CC 19 AD 1C 36
0001F8 06 1B 0B 83 08 06 84 01 C6 8E 55 60 30 08 1C 4A 10 35 A2 B0 B8 A6 80 26
000210 3F C7 36 18 11 1A 6A 3C D6 3D DB F8 DB 6F 91 BC DF 63 7D BF 8F 36 FE 3C
000228 DB B6 36 DB F8 DB 6F E3 A5 69 F2 37 9B F8 DE CF E3 6D BF 0D 37 69 BD 8D
000240 FC FE 39 6D BD 0E DB 69 23 70 DF B3 DC 61 0C 2B 05 12 4C 06 5C 7E D8 16
000258 05 94 F0 31 D6 16 B6 22 10 9A 60 42 1F 0E 52 A8 CD 32 18 84 B0 60 3E 91
000270 26 83 02 48 73 06 4E 08 2D B6 A4 2C 0E CD 8B 41 5C 03 82 03 4D 34 0C 74
000288 3A 12 C4 32 B0 63 8D 8C 5B 06 05 18 EB 01 81 6E 82 A0 60 0D 49 AA 01 20
0002A0 D0 B5 60 63 E1 60 37 0B 81 68 34 0C A4 02 5B 12 AA 06 A3 E5 65 46 04 00
0002B8 C8 C5 01 60 1B C9 D9 F0 2B 01 F2 60 09 05 A0 21 0E C2 D1 C1 E8 F1 40 24
0002D0 0A 0F 87 60 C7 DB 19 02 18 06 03 2C 08 AD 61 58 25 86 70 48 0B 60 92 13
0002E8 C1 BE 3C 69 B1 F3 6D A8 03 21 70 30 89 22 12 90 32 17 04 EC 0B 63 CC 37
000300 C0 74 2D 1F 6A 3E 46 DB 7F 1E 6D F6 36 DB F8 FB DF E3 DD EF 8E BD 34 91
000318 C3 6E B4 79 F7 F1 E6 DD B1 FD 9A F1 FD 9C 91 FA 87 F1 BD 9F C7 BB D6 C6
```

Selection: 0x46,7-0x57,0 144 bits

Ready

Setting Bytes per Line

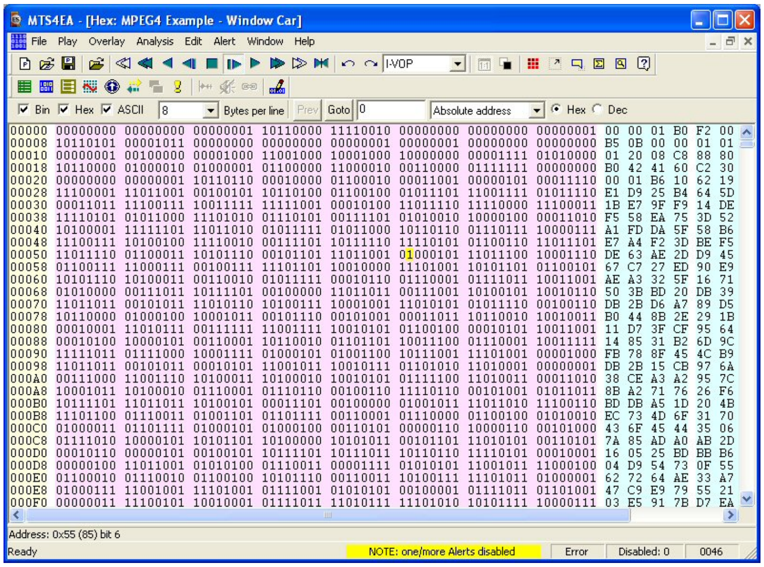
This menu is used to set the number of bytes per line:



☐ Bin ☑ Hex ☑ ASCII Auto Bytes per line Prev Goto 0

```
00000 00 00 01 B0 F2         1 B5 0B 00 00 01 01
0001B 60 C2 30 7F 00         6 10 62 19 05 1E E1
00036 F0 E3 F5 58 EA         2 84 1A A1 FD DA 5F
00051 63 AE 2D D9 45         7 C7 27 ED 90 E9 AD
0006C DB 39 A5 96 DE         7 89 D5 5E 26 B0 44
00087 99 14 85 31 B2 6D 9C 71 9F FB 78 8F 45 4C
000A2 A3 A2 95 7C D3 1A 8B A2 71 76 26 F6 29 5B
000BD 70 64 52 43 6F 45 44 35 06 86 28 7A 85 AD
000D8 04 D9 54 73 0F 55 CB C4 62 72 64 AE 33 A7
```

If the current width of the window is insufficient to show all the columns, then a scroll bar automatically appears at the bottom, allowing scrolling left/right to see all the columns, as in the following example:



☑ Bin ☑ Hex ☑ ASCII 8 Bytes per line Prev Goto 0 Absolute address ● Hex ○ Dec

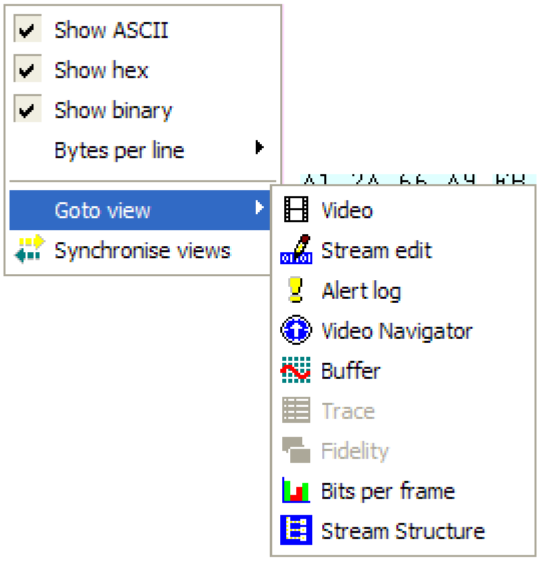
```
00000 00000000 00000000 00000001 10110000 11110010 00000000 00000000 00000001 00 00 01 B0 F2 00
00008 10110101 00001011 00000000 00000000 00000001 00000001 00000000 00000000 B5 0B 00 00 01 01
00010 00000001 00100000 00001000 11001000 10001000 10000000 00001111 01010000 01 20 08 C8 88 80
00018 10110000 01000010 01000001 01100000 11000010 00110000 01111111 00000000 B0 42 41 60 C2 30
00020 00000000 00000001 10110110 00010000 01100010 00011001 00000101 00011110 00 01 B6 10 62 19
00028 11100001 11011001 00100101 10110100 01100100 01011101 11001111 01011110 E1 D9 25 B4 64 5D
00030 00011011 11100111 10011111 11111001 00010100 11011110 11110000 11100011 1B E7 9F F9 14 DE
00038 11110101 01011000 11101010 01110101 00111101 01010010 10000100 00011010 F5 58 EA 75 3D 52
00040 10100001 11111101 11011010 01011111 01011000 10110110 01110111 10000111 A1 FD DA 5F 58 B6
00048 11100111 10100100 11110010 00111101 10111110 11110101 01100110 11011101 E7 A4 F2 3D BE F5
00050 11011110 01100011 10101110 00101101 11011001 01000101 11011100 10001110 DE 63 AE 2D D9 45
00058 01100111 11000111 00100111 11101101 10010000 11101001 10101101 01100101 67 C7 27 ED 90 E9
00060 10101110 10100011 00110010 01011111 00010110 01110001 01111011 10011001 AE A3 32 5F 16 71
00068 01010000 00111011 10111101 00100000 11011011 00111001 10100101 10010110 50 3B BD 20 DB 39
00070 11011011 00101011 11010110 10100111 10001001 11010101 01011110 00100110 DB 2B D6 A7 89 D5
00078 10110000 01000100 10001011 00101110 00101001 00011011 10110010 10010011 B0 44 8B 2E 29 1B
00080 00010001 11010111 00111111 11001111 10010101 01100100 00010101 10011001 11 D7 3F CF 95 64
00088 00010100 10000101 00110001 10110010 01101101 10011100 01110001 10011111 14 85 31 B2 6D 9C
00090 11111011 01111000 10001111 01000101 01001100 10111001 11101001 00001000 FB 78 8F 45 4C B9
00098 11011011 00101011 00010101 11001011 10010111 01101010 11010001 00000001 DB 2B 15 CB 97 6A
000A0 00111000 11001110 10100011 10100010 10010101 01111100 11010011 00011010 38 CE A3 A2 95 7C
000A8 10001011 10100010 01110001 01110110 00100110 11110110 00101001 01011011 8B A2 71 76 26 F6
000B0 10111101 11011011 10100101 00011101 00100000 01001011 11011010 11100110 BD DB A5 1D 20 4B
000B8 11101100 01110011 01001101 01101111 00110001 01110000 01100100 01010010 EC 73 4D 6F 31 70
000C0 01000011 01101111 01000101 01000100 00110101 00000110 10000110 00101000 43 6F 45 44 35 06
000C8 01111010 10000101 10101101 10100000 10101011 00101101 11010101 00110101 7A 85 AD A0 AB 2D
000D0 00010110 00000101 00100101 10111101 10111011 10110110 11110101 00010001 16 05 25 BD BB B6
000D8 00000100 11011001 01010100 01110011 00001111 01010101 11001011 11000100 04 D9 54 73 0F 55
000E0 01100010 01110010 01100100 10101110 00110011 10100111 11101011 01000001 62 72 64 AE 33 A7
000E8 01000111 11001001 11101001 01111001 01010101 00100001 01111011 01101001 47 C9 E9 79 55 21
000F0 00000011 11100101 10010001 01111011 11010111 11101010 10101111 10000111 03 E5 91 7B D7 EA
```

Address: 0x55 (85) bit 6

Ready NOTE: one/more Alerts disabled Error Disabled: 0 0046

Right-Click Popup Menu/Goto View

Right-clicking in the HexView window generates the menu:



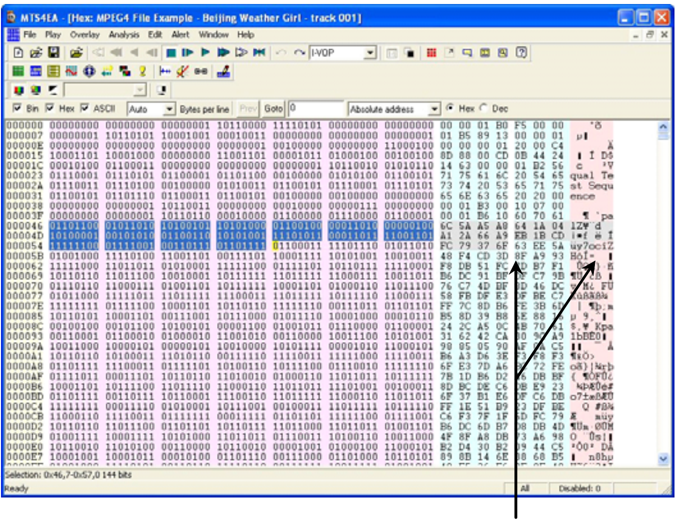
The top four buttons of the menu have the same functions as the buttons given on the top line (and as the HexView settings... option on the Window menu).

Goto view. The Goto view takes the focus to the corresponding area in the selected view (see *Synchronized Views/Navigating the Views* on page 6–7 for more information).

Synchronize views. When the Synchronize views icon ⇄ is pushed in, then all open windows automatically follow the selection (see *Synchronized Views/Navigating the Views* on page 6–7 for more information).

Highlighting a Section

A section of data can be highlighted in any of the (visible) binary, hex or ASCII sections by clicking and dragging the mouse over the area:



MTS4EA - [Hex: MPEG4 File Example - Beijing Weather Girl - track 001]

Selection: 0x46,7-0x57,0 144 bits

Ready

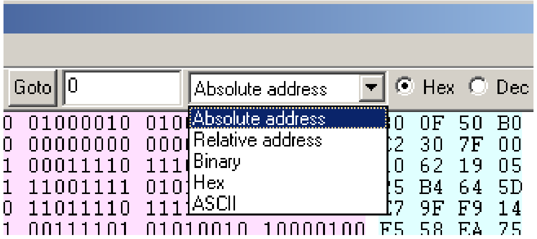
The addresses of the selected area and the number of bytes selected are displayed in the status line

Whichever section is highlighted, the corresponding areas of the other sections are similarly highlighted

A specific address can be found in the video stream, either of the following:

- An absolute address, from the start of the video file
- A relative address, from the first byte of the currently selected area

Find absolute address. To go to an absolute address, select Absolute address from the drop-down menu:



The address to find is entered in the box next to the Goto button. This address can be in hex or decimal. If an invalid character is entered for an address (for example, entering anything other than 0-9 or a-f for a hexadecimal address) then the Goto button is grayed out.

Note that the Goto button changes to Next for all data to find except for the Absolute address. (And when Absolute address is selected, the Prev button is grayed out.)

The Hex/Dec radio button selects the format of the data being searched - Hexademical or Decimal.

Find relative address. An address relative to the currently highlighted address can be found. If no address is highlighted, the address found is the offset from 0.

The address to find is entered in the box next to the Next button. This address can be in hex or decimal. If an invalid character is entered for an address (for example, entering anything other than 0-9 or a-f for a hexadecimal address), the Find button is grayed out.

NOTE. *The F3 key can be used, to find next; Shift+F3 key, to find previous.*

The Hex/Dec radio button selects the format of the data being searched.

Find Binary/Hex/ASCII. These options in the menu find data in the bitstream.

Up to 64 characters can be entered.

NOTE. *The Binary search searches for the bit pattern regardless of byte location; the Hex and ASCII searches are byte aligned.*

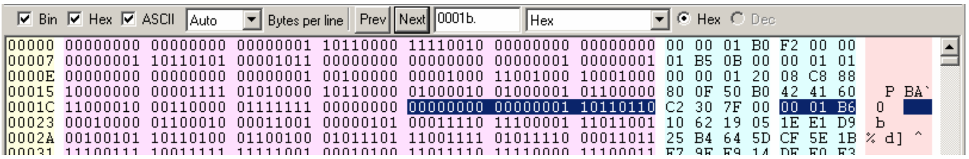
The F3 key can be used, to find next; Shift+F3 key, to find previous.

Wildcard searching using period. A wildcard can be entered in the data to be found - **.** [period].

The wildcard matches a single digit in the selected base; the wildcard matches:

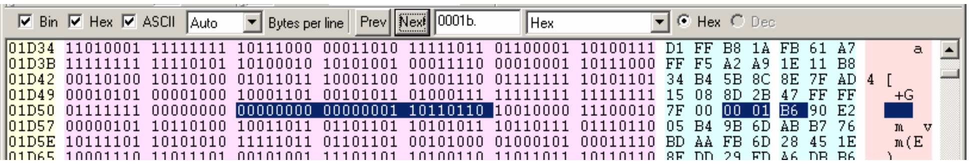
- One bit in the binary find
- A hex digit in the hex find
- An ASCII character in the ASCII find

For example, entering ‘0001B.’ as a hex string will find the following patterns in the selected example stream:



```
00000 00000000 00000000 00000001 10110000 11110010 00000000 00000000 00 00 01 B0 F2 00 00
00007 00000001 10110101 00001011 00000000 00000000 00000001 00000001 01 B5 0B 00 00 01 01
0000E 00000000 00000000 00000001 00100000 00001000 11001000 10001000 00 00 01 20 08 C8 88
00015 10000000 00001111 01010000 10110000 01000010 01000001 01100000 80 0F 50 B0 42 41 60   P BA`
0001C 11000010 00110000 01111111 00000000 00000000 00000001 10110110 C2 30 7F 00 00 01 B6  0
00023 00010000 01100010 00011001 00000101 00011110 11100001 11011001 10 62 19 05 1E E1 D9  b
0002A 00100101 10110100 01100100 01011101 11001111 01011110 00011011 25 B4 64 5D CF 5E 1B % d] ^
00031 11100111 10011111 11111001 00010100 11011110 11110000 11100011 E7 9F F9 14 DE F0 E3
```

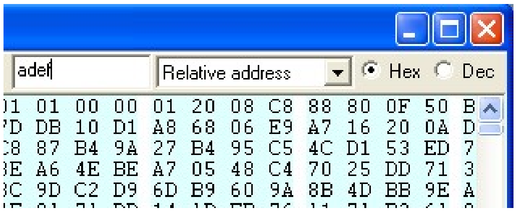
The wildcard can be inserted in any position, for example, ‘03..456.8’ is a valid search string, which will find any 9 consecutive digits where the first two are 03, digits 5-7 are 456, and the last digit is 8.



```
01D34 11010001 11111111 10111000 00011010 11111011 01100001 10100111 D1 FF B8 1A FB 61 A7     a
01D3B 11111111 11110101 10100010 10101001 00011110 00010001 10111000 FF F5 A2 A9 1E 11 B8
01D42 00110100 10110100 01011011 10001100 10001110 01111111 10101101 34 B4 5B 8C 8E 7F AD 4 [
01D49 00010101 00001000 10001101 00101011 01000111 11111111 11111111 15 08 8D 2B 47 FF FF   +G
01D50 01111111 00000000 00000000 00000001 10110110 10010000 11100010 7F 00 00 01 B6 90 E2
01D57 00000101 10110100 10011011 01101101 10101011 10110111 01110110 05 B4 9B 6D AB B7 76   m  v
01D5E 10111101 10101010 11111011 01101101 00101000 01000101 00011110 BD AA FB 6D 28 45 1E   m(E
01D65 10001110 11011101 00101001 11101101 10100110 11011011 10110110 8E DD 29 ED A6 DB B6   )
```

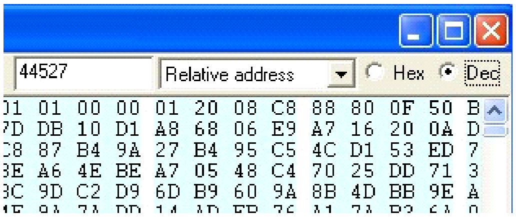
Conversion of Hex<->Decimal

If a hex value is entered into the find box:



```
)1 01 00 00 01 20 08 C8 88 80 0F 50 B
'D DB 10 D1 A8 68 06 E9 A7 16 20 0A D
:8 87 B4 9A 27 B4 95 C5 4C D1 53 ED 7
3E A6 4E BE A7 05 48 C4 70 25 DD 71 3
3C 9D C2 D9 6D B9 60 9A 8B 4D BB 9E A
```

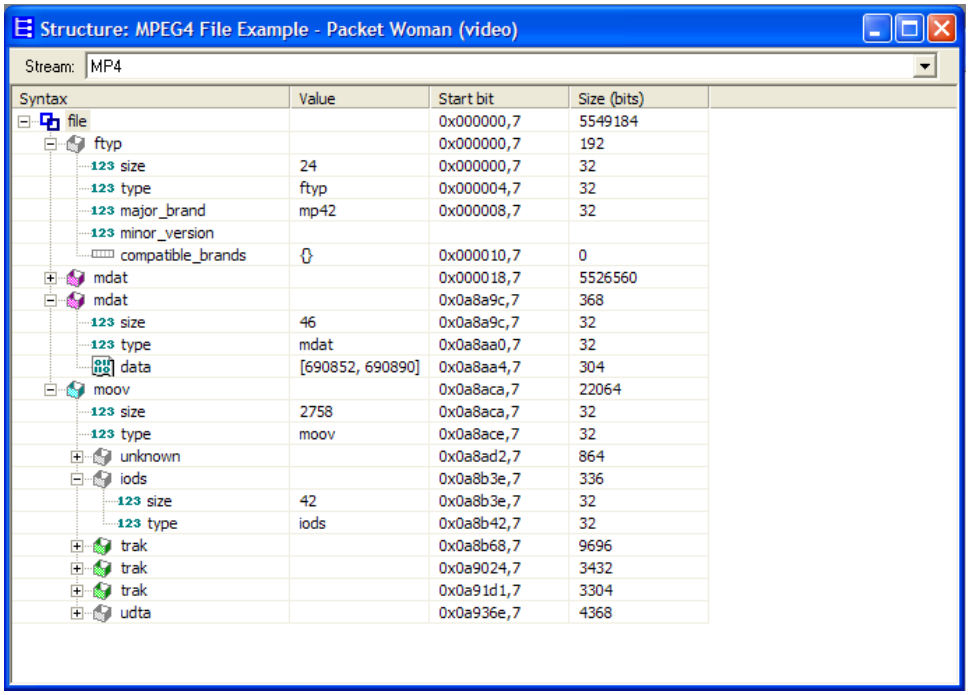
When the Dec button is clicked, this value is converted to hex (and vice versa):



```
)1 01 00 00 01 20 08 C8 88 80 0F 50 B
'D DB 10 D1 A8 68 06 E9 A7 16 20 0A D
:8 87 B4 9A 27 B4 95 C5 4C D1 53 ED 7
3E A6 4E BE A7 05 48 C4 70 25 DD 71 3
3C 9D C2 D9 6D B9 60 9A 8B 4D BB 9E A
```

View Video Stream Structure... Ctrl+R

This allows the structure of all supported container formats to be viewed and expanded/contracted (by clicking on the - and + symbols at the left end of each line). In addition, for H.264/AVC video elementary stream format, the view shows the top-level structure of the stream.



Structure: MPEG4 File Example - Packet Woman (video)

Stream: MP4

Syntax	Value	Start bit	Size (bits)
⊟ file		0x000000,7	5549184
⊟ ftyp		0x000000,7	192
123 size	24	0x000000,7	32
123 type	ftyp	0x000004,7	32
123 major_brand	mp42	0x000008,7	32
123 minor_version			
compatible_brands	{}	0x000010,7	0
⊞ mdat		0x000018,7	5526560
⊟ mdat		0x0a8a9c,7	368
123 size	46	0x0a8a9c,7	32
123 type	mdat	0x0a8aa0,7	32
data	[690852, 690890]	0x0a8aa4,7	304
⊟ moov		0x0a8aca,7	22064
123 size	2758	0x0a8aca,7	32
123 type	moov	0x0a8ace,7	32
⊞ unknown		0x0a8ad2,7	864
⊟ iods		0x0a8b3e,7	336
123 size	42	0x0a8b3e,7	32
123 type	iods	0x0a8b42,7	32
⊞ trak		0x0a8b68,7	9696
⊞ trak		0x0a9024,7	3432
⊞ trak		0x0a91d1,7	3304
⊞ udta		0x0a936e,7	4368

The view is divided into four columns:

- **Syntax**: Shows the stream syntax parse tree, including the names of each syntax element or syntactic structure. Nested syntax structures are shown by nesting nodes of the tree. An icon shows the type of data at each tree node, see *Icons in Stream Structure View*.
- **Value**: Shows the interpreted value of syntax elements in the tree. The content of this field will vary according to the type of data to be shown.
- **Start bit**: Shows the start bit address of the syntax node. The address is given as a hexadecimal byte position followed by a bit position number.
- **Size**: Shows the size in bits of the syntax node. For nodes with children, this shows the total bits of the node and all its children.

Icons in Stream Structure View

In the Stream structure view, the icons that appear at the node of each branch of the tree structure denote the type of data contained in that node. There are a number of icons which are common to all standards as described in Table 19-4. Table 19-5 show icons that are specific to different standards.

Table 19-4: Icons general to all standards

Icon	Meaning
●	Video elementary stream
((•))	Audio elementary stream
⧉	Container stream
🗀	Packet or other syntactic grouping (various colors used – see standard specific icons)
01 110	Binary data
123	Integer data field
101	Binary data field
01X	Exponential-Golomb coded data field
ab	String data field
{ID}	Identifier data field
⏱	Time data field
◇	Custom data field
?	Unknown data field
☺	User data
▭	Array based data structure
⧉	List based data structure
▦	Video ES slice data
⊞	Video ES macroblock data

Table 19-5: Standard specific icons

Standard	Icon	Meaning
H.264/AVC	🗀	IDR access unit
	🗀	Non-IDR access unit
	📦	‘moov’ box type
	📦	‘mdat’ box type
	📦	‘trak’ box type
	🗀	Generic MP4 descriptor
MPEG-2 TS	🗀	Transport packet
	🗀	Program association section
MPEG-2 PS, PES	🗀	System header
	🗀	Pack header
	🗀	Pack syntactic structure
ASF	🗀	Header object
	🗀	File properties object
	🗀	Stream properties object
	🗀	Index object

View Video Buffer Analysis...

This allows the buffer usage to be analyzed in detail. The analysis varies by video standard:

For H.264/AVC the analysis is of:

- Performance using the Hypothetical Reference Decoder (HRD) (displayed in red)

For MPEG-4 the analysis is of:

- Video Buffer Verifier (displayed in Red)
- Video Complexity Verifier (displayed in Green)
- Video Memory Verifier (displayed in Blue)

For MPEG-2 the analysis is of:

- Video Buffer Verifier (displayed in Red)

The icon controls and general look of the buffer analysis is the same for all standards:

- The buffer analysis controls that are common to all standards are explained in the sections beginning *Buffer Analysis Toolbar Icons* on page 6–187 to *Buffer analysis alerts/scroll bar area* on page 6–190 (although some of the diagrams in these sections are specific to a standard regarding the titles and data content, the functions are the same across all standards)

- MPEG-4 and MPEG-2 VBV analysis are very similar and are explained in the sections beginning *Buffer analysis controls: MPEG-4 and MPEG-2* on page 6–191 to *Buffer analysis popup alerts: MPEG-4 and MPEG-2* on page 6–193

- The H.264/AVC HRD analysis is distinct and is explained in the sections beginning *HRD buffer analysis: H.264/AVC* on page 6–193 to *HRD buffer overflow/underflow indication: H.264/AVC* on page 6–195

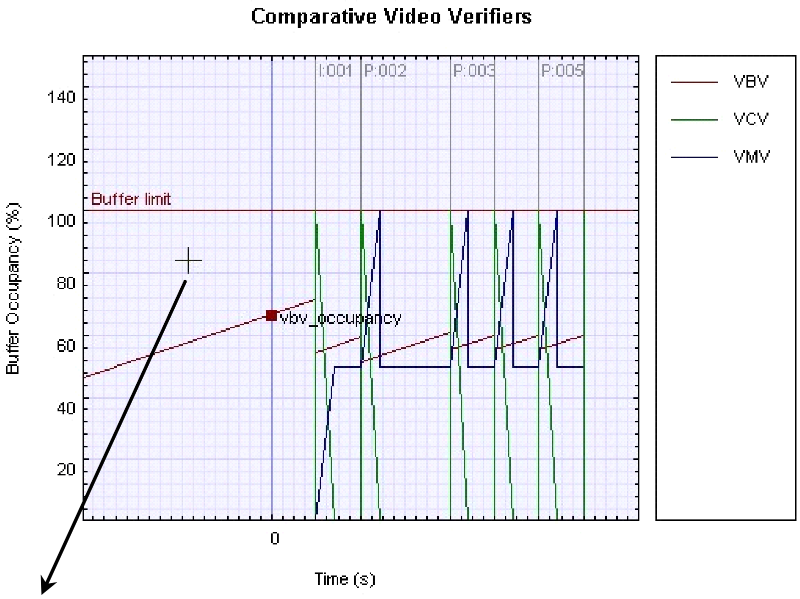
Buffer Analysis Toolbar Icons

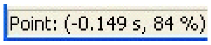
Icon	Function
+ −	Zoom in (+) and zoom out (-) centered on the center of the window (affects scale of x-axis only)
🗋	Fit all data into window
○	Locate origin (zero), start of sequence
🔒X 🔒Y	Lock X/Y zoom in/out and scrolling/panning. For example, when the Lock Y button is pressed, zoom in and zoom out and scroll/pan only affect the X-direction. This allows (for example) you to keep a useful vertical scale, while still viewing the whole length of the video sequence
→	Autoscroll (fill the analysis data in real-time) as the video is being decoded and scroll the window to the right
↕+	Increase track height (affects scale of y-axis only)
↕−	Decrease track height (affects scale of y-axis only)
+	Measure the data at the cursor. The data values are reported on the status line at the bottom of the analysis window Offsets and angles/slopes of lines can also be measured, by holding the mouse and dragging
✥	Scroll/pan (the cursor changes to show the scroll/pan direction)
🔍	Zoom in/zoom out centered on the location of this cursor. Press the <shift> key to zoom out

These functions are also available on the right-click menu.

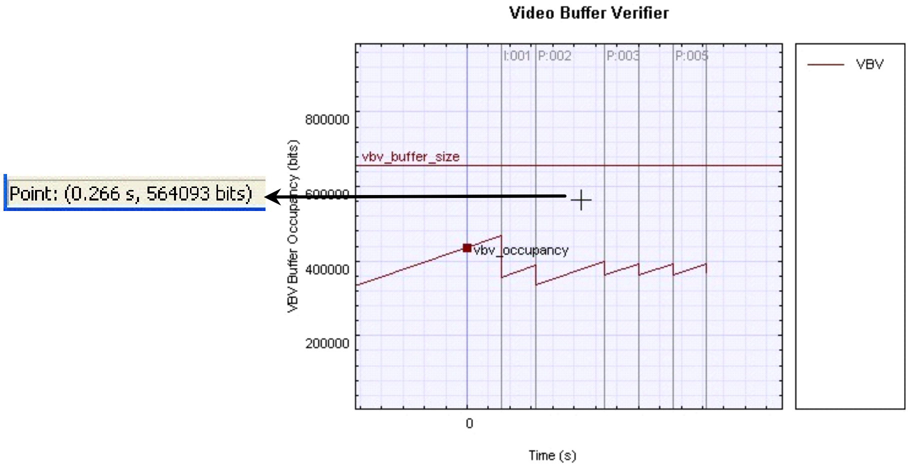
Using the Measure Tool +

Moving the measure tool over the graph displays the values in the status line:

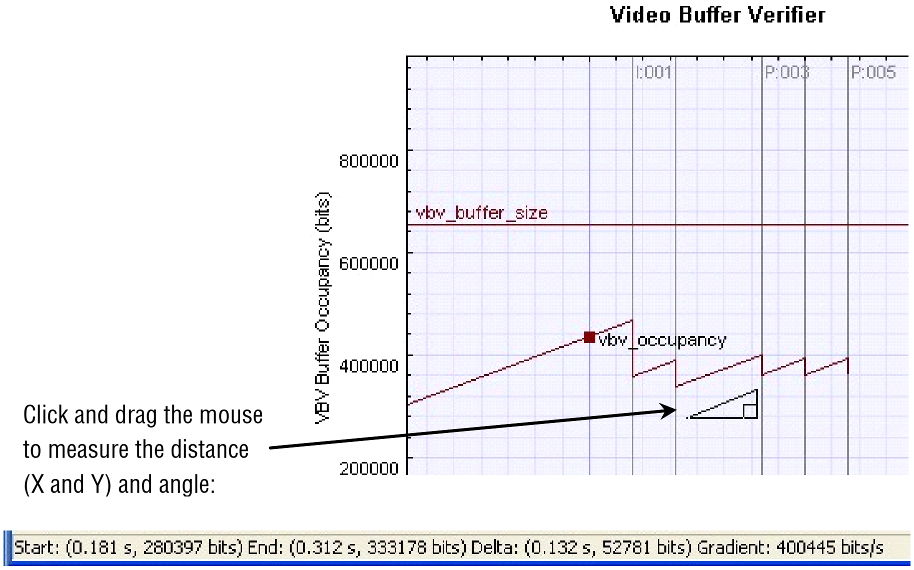



Point: (-0.149 s, 84 %)

The left-hand value is the X-axis (time) value in seconds, to the nearest millisecond; the right-hand is the Y-axis value: either in percentage terms, if there is more than one graph; or in the correct units, if there is only one graph (as below)


Point: (0.266 s, 564093 bits)

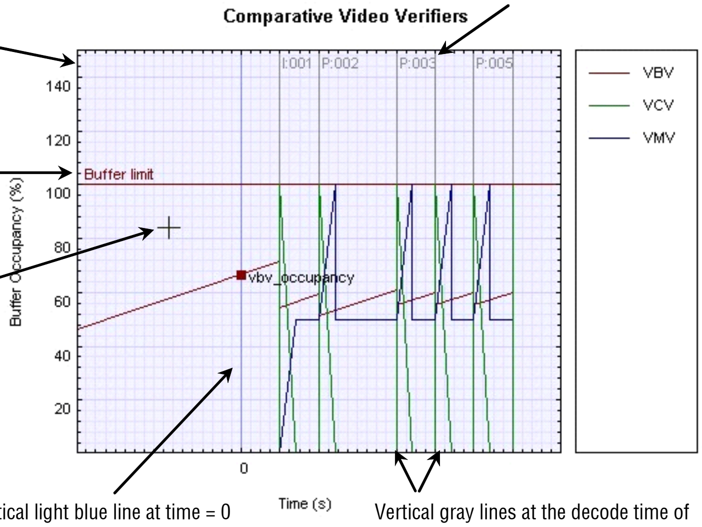
(Only the VBV is shown here)



Click and drag the mouse to measure the distance (X and Y) and angle:

Start: (0.181 s, 280397 bits) End: (0.312 s, 333178 bits) Delta: (0.132 s, 52781 bits) Gradient: 400445 bits/s

Buffer Analysis Graph Area



Scale - either normalized to 100%, if more than one graph, or else in actual units

Buffer limit (100%)

Measure tool with X and Y values reported in the status

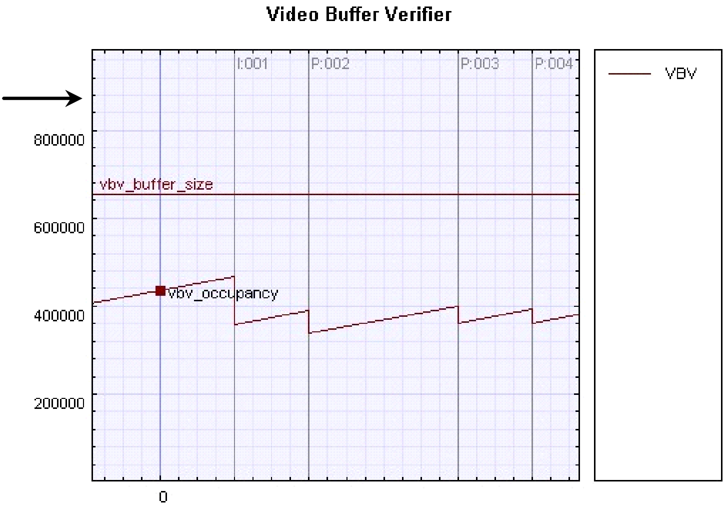
Frame type and number

Vertical light blue line at time = 0

Vertical gray lines at the decode time of each frame

The left axis displays:

- Values normalized to 100%, if there is more than one graph
- Values appropriate for that graph (see the following figure)



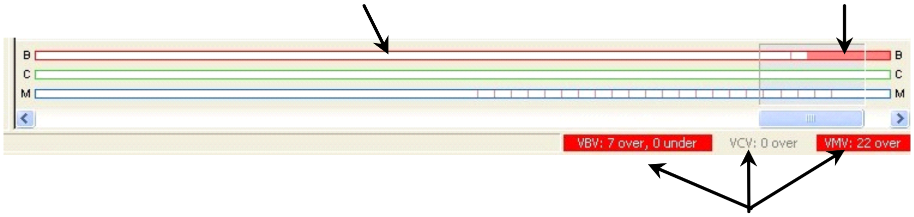
Vertical scale now in units correct for VBV buffer size (in bits)

When the buffer data is too wide/too high for the current window, scroll bars appear at the bottom/right (as appropriate).

Buffer analysis alerts/scroll bar area. The buffer overflow/underflow is indicated in the Graph window as shown below:

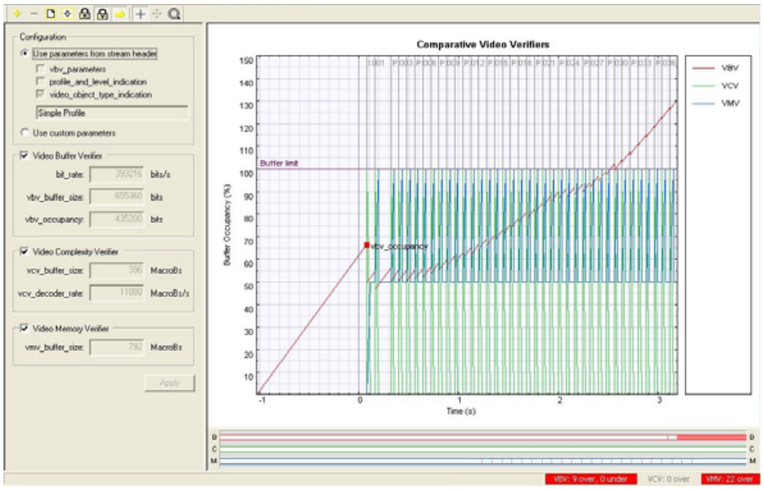
Each bar line fills and auto-scales, corresponding with the horizontal scroll bar (B=VBV, C=VCV, M=VMV)

Red shows overflow; yellow shows underflow

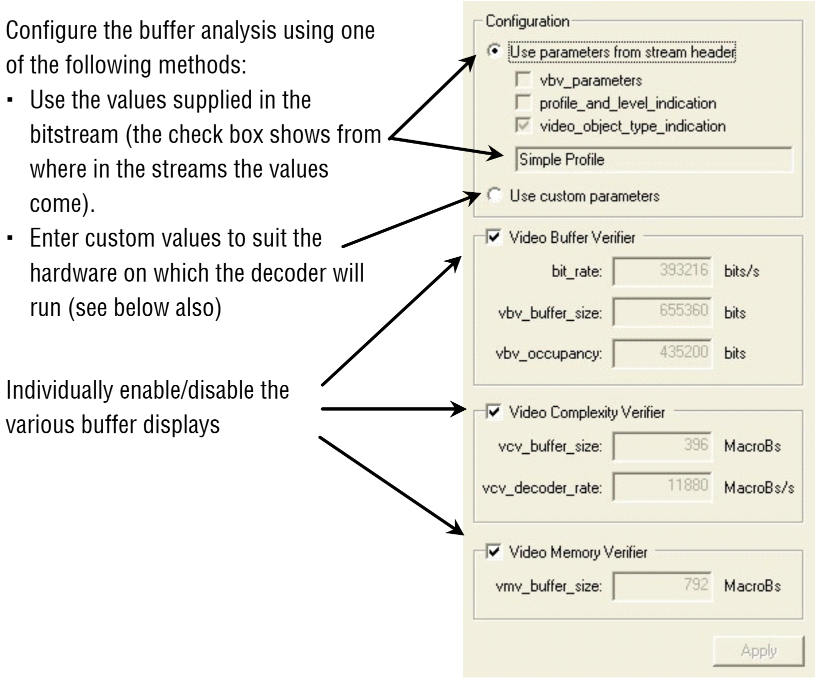


Each buffer item changes to red or yellow if there has been any overflow/underflow, and the number indicates in how many frames there is non-conformance. If there is both overflow and underflow, the color is red.

MPEG-4 and MPEG-2 buffer analysis. All the example screenshots and explanations below are for MPEG-4; however, similar displays and information are used for MPEG-2.



Buffer analysis controls: MPEG-4 and MPEG-2.



Configure the buffer analysis using one of the following methods:

- Use the values supplied in the bitstream (the check box shows from where in the streams the values come).
- Enter custom values to suit the hardware on which the decoder will run (see below also)

Individually enable/disable the various buffer displays

Configuration

● Use parameters from stream header

□ vbv_parameters

□ profile_and_level_indication

☑ video_object_type_indication

Simple Profile

○ Use custom parameters

☑ Video Buffer Verifier

bit_rate: 393216 bits/s

vbv_buffer_size: 655360 bits

vbv_occupancy: 435200 bits

☑ Video Complexity Verifier

vcv_buffer_size: 396 MacroBs

vcv_decoder_rate: 11880 MacroBs/s

☑ Video Memory Verifier

vmv_buffer_size: 792 MacroBs

The values used for VBV, VCV and VMV are displayed, but cannot be altered unless Use custom parameters is selected.

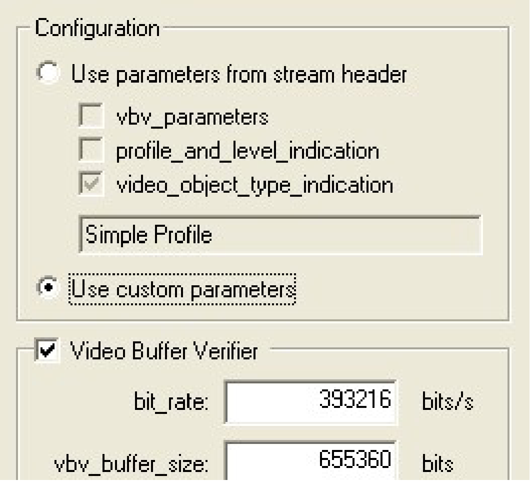
The boxes below the title Use parameters from stream header show where the values used have come from: in the example above, there were no values specified in the `vbv_parameters`, nor was there a `profile_and_level_indication`, so the maximum value implied by Simple Profile as given in the `video_object_type_indication` is used.

NOTE. *There is an order of precedence: if the `vbv_parameters` are specified in the bitstream, then these values are used; if not, then the maximum values are used as allowed in the Profile/Level given by the `profile_and_level_indication`. If there is no `profile_and_level_indication`, then the maximum values are used as implied by the `video_object_type_indication`.*

Use custom buffer parameters: MPEG-4 and MPEG-2. Custom parameters can be entered for a variety of reasons, for example:

- The VBV/VCV/VMV parameters have not been specified in the bitstream: in this case the MTS4EA will assume the maximum allowable values for the Profile/Level and these may be too large for the particular operation.
- The VBV/VCV/VMV parameters specified in the bitstream do not reflect the actual limitations of the hardware on which the decoder will have to decode the bitstream.
- To try different values to see if the conformance parameters are met with different limits.

Custom parameters are entered as below:



Configuration

○ Use parameters from stream header

□ vbv_parameters

□ profile_and_level_indication

☑ video_object_type_indication

Simple Profile

● Use custom parameters

☑ Video Buffer Verifier

bit_rate: 393216 bits/s

vbv_buffer_size: 655360 bits

When custom parameters are used, this is indicated in the status bar of the main MTS4EA window by the word BUF:

NOTE. *These values can also be selected/entered from the Decoder options selection on the Play menu (the MPEG-4 tab).*

The values entered as Custom parameters affect both the alert popup warnings and the warnings that are reported in the Trace files; the Custom parameters are the values used to trigger these warnings.

To reset the values entered to the values specified by the bitstream, reselect the Use parameters button from the stream header.

Buffer analysis popup alerts: MPEG-4 and MPEG-2. Where the bitstream exceeds the values given by:

- `vbv_parameters`
- `profile_and_level_indication`
- The custom parameters

When each of these is selected as the source of the parameters, popup alerts occur during video decoding unless disabled in the Alerts menu (the video must be playing for the popup alerts to be generated).

NOTE. *When the buffer parameters are implied by* `video_object_type_indication`*, no popup alert is generated in the event of exceeding the limits, nor is there a warning in the Trace files. This is in accordance with the implied rules in the MPEG-4 standard.*

HRD buffer analysis: H.264/AVC. The Hypothetical Reference Decoder (HRD) in H.264/AVC provides a complex mechanism for buffer analysis; this is much more complex than the buffer analysis in MPEG-4/MPEG-2.

NOTE. *The HRD analysis is of the Coded Picture Buffer (CPB) only, not of the Decoded Picture Buffer (DPB).*

In order to understand the HRD analysis, you must read and understand Annex C of the H.264/AVC standard, ISO document 14496-10 (E).

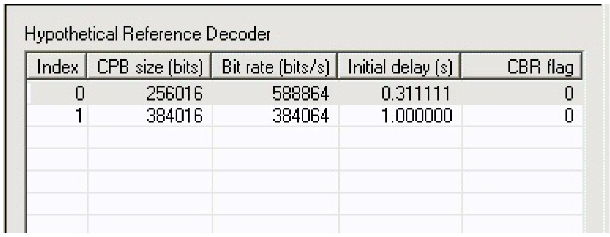
HRD parameters do not have to be specified in a bitstream; many H.264/AVC streams do not have them specified (only the Bus Junction and Grenadier Guards H.264/AVC example streams provided with the MTS4EA have HRD information).

NOTE. *If HRD parameters are not provided in an H.264/AVC bitstream, the icon and menu item for View buffer analysis... is grayed out.*

HRD list of schedule indexes (buckets): H.264/AVC. Due to the complexity of HRD analysis, entering custom parameters is not currently possible in MTS4EA; the display shows the list of indexes or buckets that are given in the bitstream.

If the HRD parameters are given in a bitstream, there can be 1 to 31 indexes or buckets, numbered from 0.

In the following example below, there are two indexes/buckets: index 0 is selected and is used for the graph display, as shown. Clicking on the line below would display the data from index 1 in the graph display.



Hypothetical Reference Decoder

Index	CPB size (bits)	Bit rate (bits/s)	Initial delay (s)	CBR flag
0	256016	588864	0.311111	0
1	384016	384064	1.000000	0

The meanings of the column headings are:

Index: The bucket or index number.

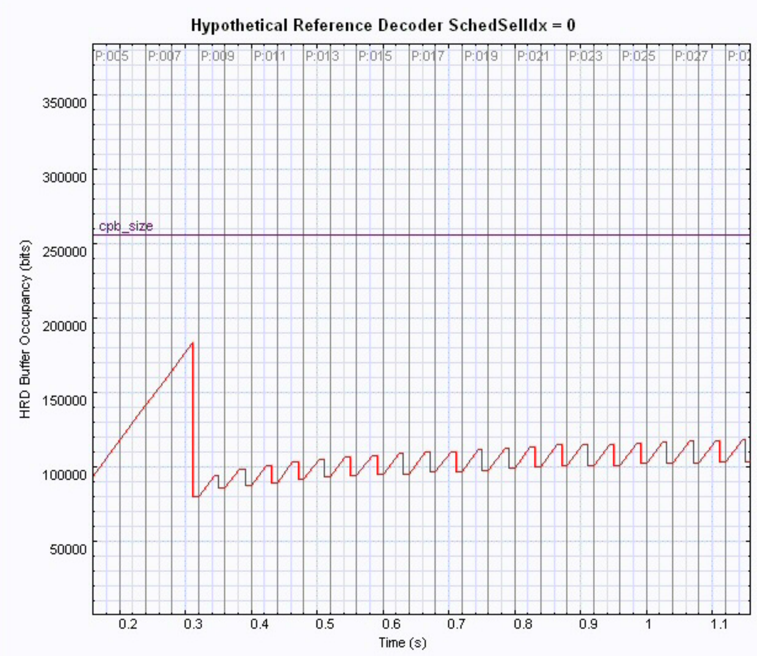
CPB size (bits): The size in bits of the Coded Picture Buffer (CPB).

Bit rate (bits/s): The bit rate of the hypothetical transmission channel.

Initial delay (s): The initial delay of the hypothetical transmission channel before the first frame is decoded.

CBR flag: The status of the Constant Bit Rate (CBR) flag: 0=off; 1 = on.

HRD results display: H.264/AVC. When a bitstream has HRD parameters specified, and the HRD buffer analysis has been done by MTS4EA, clicking on the appropriate index/bucket number displays the graph for that index/bucket (in the following example, index/bucket number 0):



The areas of the graph display are explained in *Buffer Analysis Graph Area* on page 6–189 and *Buffer analysis alerts/scroll bar area* on page 6–190.

As examples:

- Frame type and number is given at the top of the graph (in light gray)
- The vertical lines correspond with the decode times of the relevant frame

HRD buffer overflow/underflow indication: H.264/AVC. HRD buffer analysis overflow and underflow is indicated in the following locations:

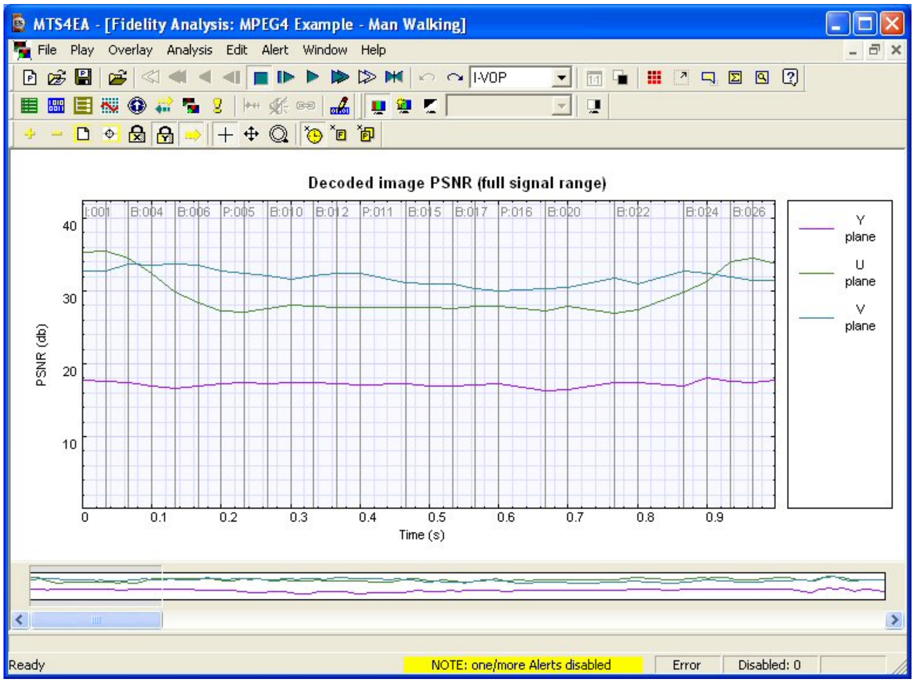
- The bar line below the graph area
- The status bar
- Popup alerts

The bar line and status bar indications are the same as overflow and underflow indications for MPEG-4/MPEG-2 (see *Buffer analysis alerts/scroll bar area* on page 6–190).

The popup alerts for HRD buffer overflow/underflow are shown on-screen and enabled/disabled as with any other popup alert (see *Explanation of Fatal/Error/Warning/Info Display* on page 6–239).

View Video Fidelity Analysis...

This displays the fidelity analysis graph view:



The fidelity analysis graph view is similar to the graph view used for buffer analysis in MPEG-4, MPEG-2 and H.264/AVC. Both views have the same:

- Control icons – see *Buffer Analysis Toolbar Icons* on page 6–187
- Method of displaying frame types, frame numbers, frame times in the graph area - see *Buffer Analysis Graph Area* on page 6–189
- Means of measuring angles – see *Using the Measure Tool* + on page 6–188
- Right-click popup menu

The fidelity analysis results are shown separately for Y, U and V planes:

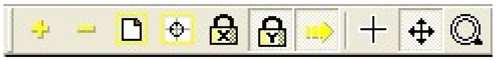
- Y-plane in mauve
- U-plane in green
- V-plane in blue

NOTE. *The MPEG-4 example stream Man Walking has the YUV reference file supplied with MTS4EA: therefore this is a useful example to choose to examine the fidelity analysis functions.*

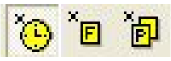
The first ten frames of the YUV reference file are supplied for the Grenadier Guards example streams that are provided with MTS4EA both as an H.264/AVC example and as an MPEG-2 example.

Where the frame rate is incorrectly set for the uncompressed video file, then the correspondence between the encoded video frames and the uncompressed video source frames is lost; this will substantially reduce the fidelity analysis values.

Fidelity Analysis View Icons Toolbar



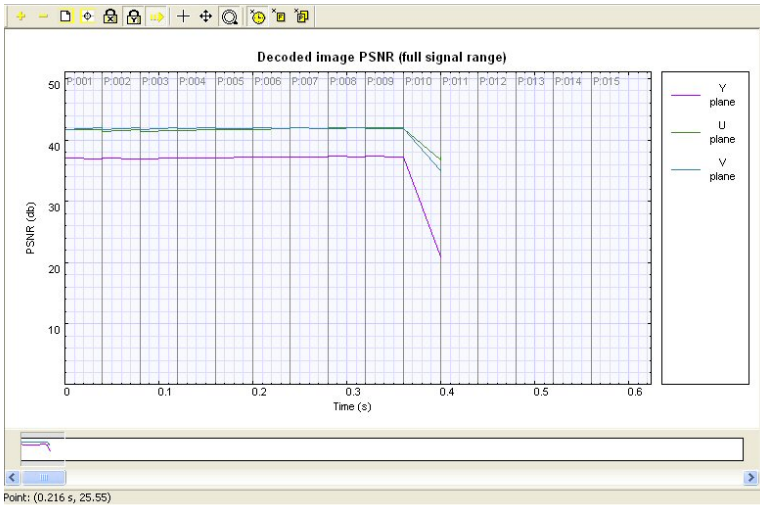
The icons above have the same functions as the icons on the buffer analysis toolbar; see *Buffer Analysis Toolbar Icons* on page 6–187. The icons below are unique to the Fidelity analysis view:



Of these three, the first two are available at all times. The third is only available when interlacing is in use and the picture can be broken down into units; it is grayed out at all other times.

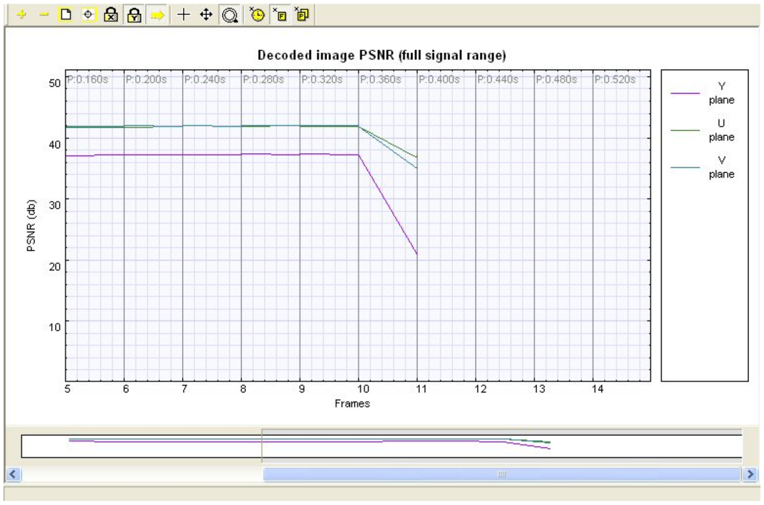
These icons represent the following:

- The ability to plot time in seconds on the x-axis, as shown:

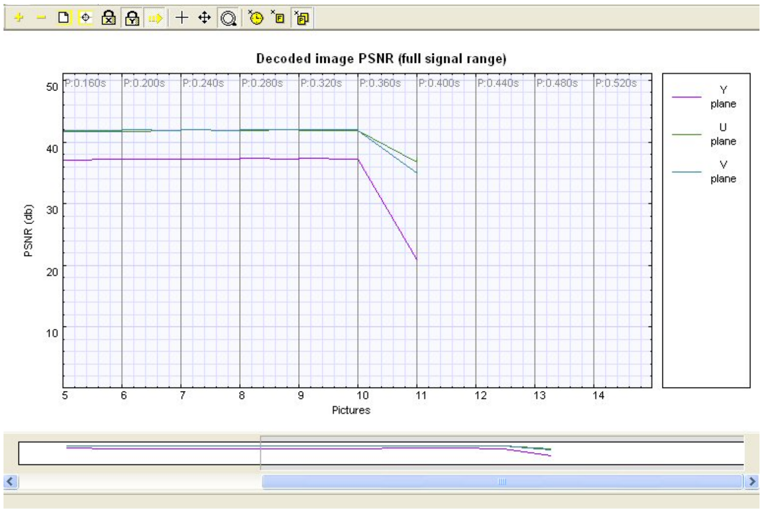


Point: (0.216 s, 25.55)

- The ability to plot frame number on the x-axis, as shown:



- The ability to plot picture units on the x-axis, as shown:



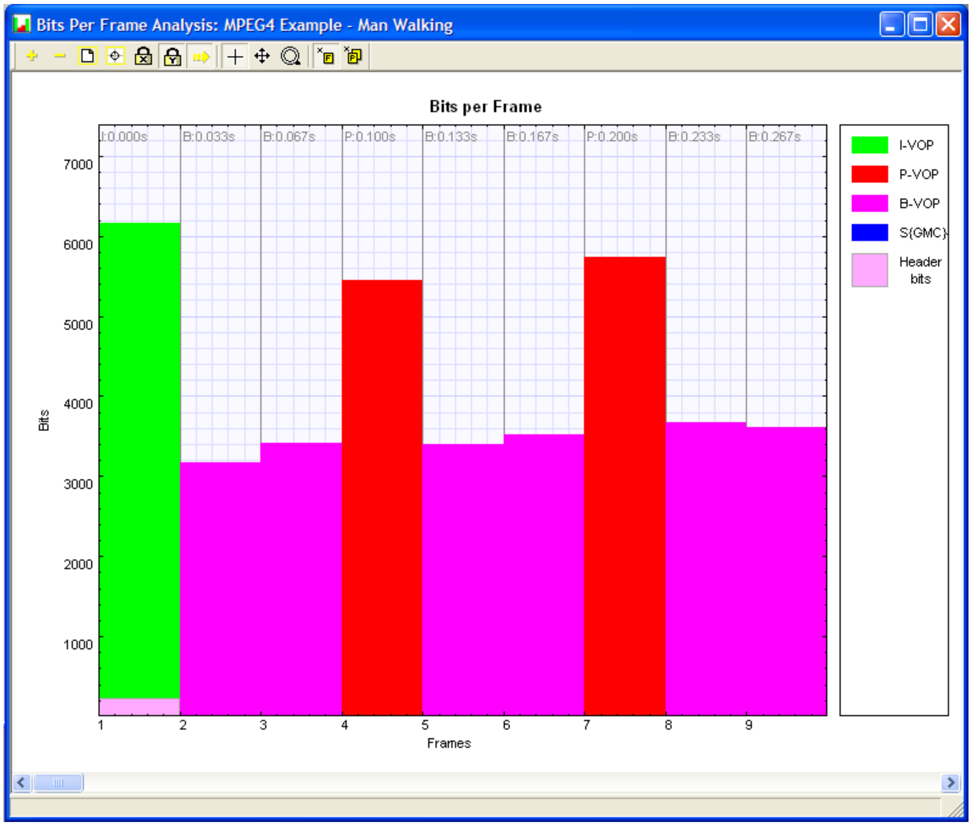
Fidelity metrics available. Six different fidelity metrics are available:

- PSNR (255 signal range)
- PSNR (ITU-R BT.601 signal range)
- RMSE (Root Mean Square Error)
- MSE (Mean Square Error)
- MAD (Mean Absolute Difference)
- SAD (Sum Absolute Difference)

These are described in detail under *Metric* on page 6–165.

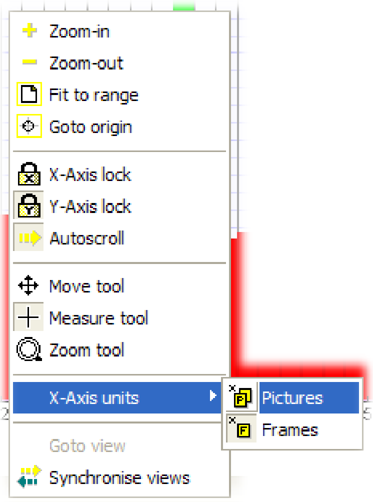
View Bits per Video Frame Analysis...

This displays the Bits per Frame Analysis graph.

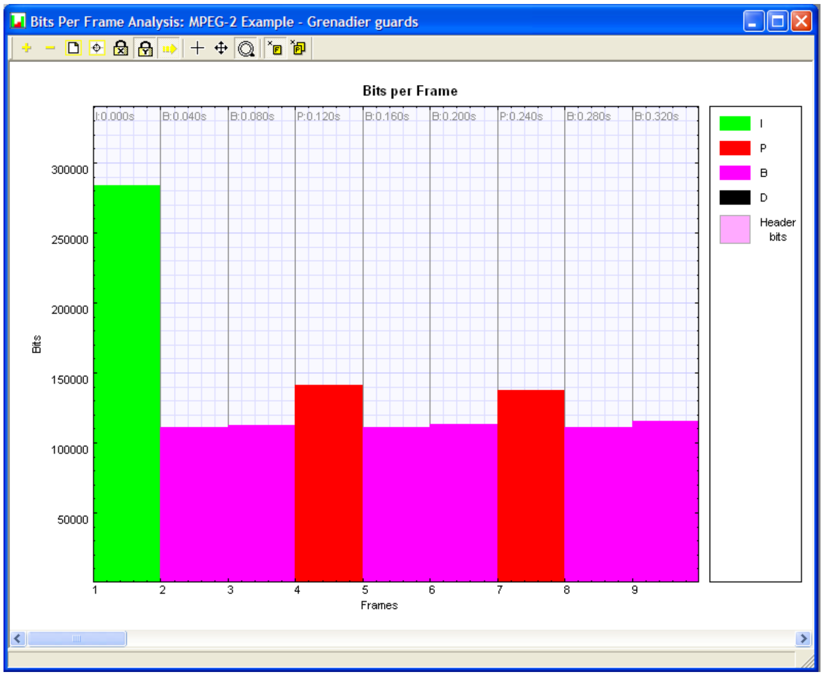


The key (on the right side) gives a color code for the VOP/frame types; for more information see *VOP type* [MPEG-4] or Frame type [H.264/AVC, MPEG-2, H.263, VC-1]. on page 6–79.

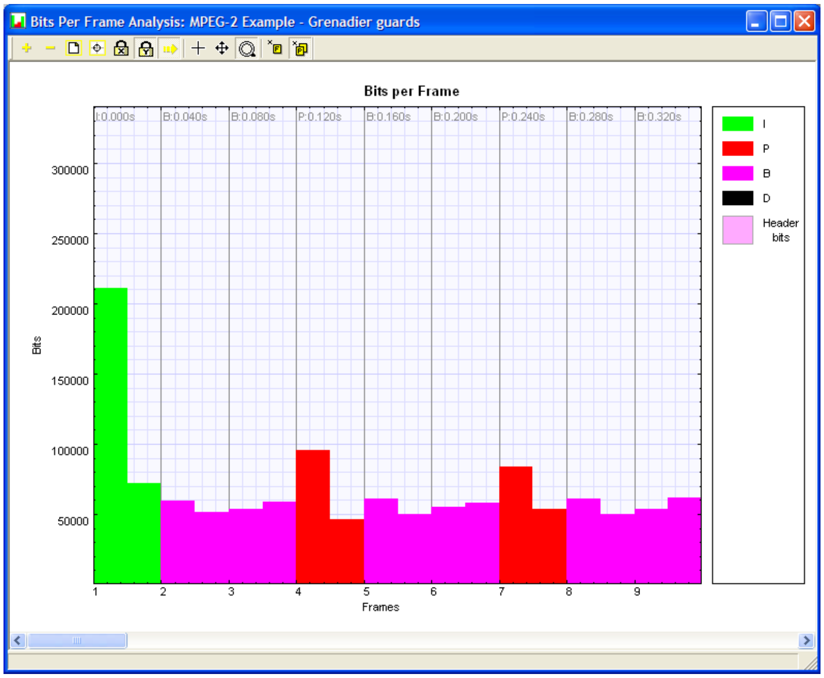
In addition to the standard buttons, the toolbar contains two buttons that allow the X-axis to be either frame or picture oriented:



The Frame unit toolbar button displays the total bits per frame; for interlaced pictures this would be the sum of the picture fields:



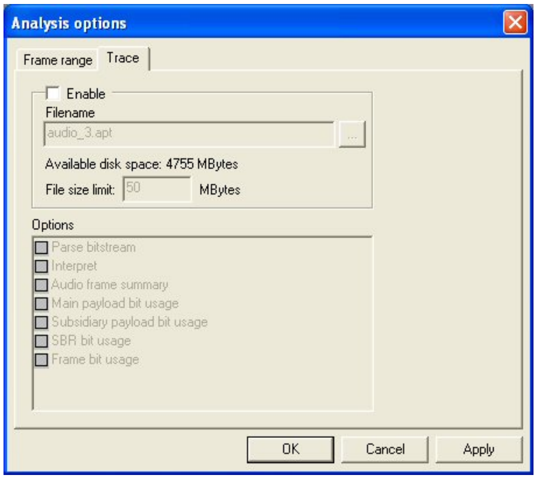
The Picture units toolbar button also displays the total bits per frame, but, in the case of interlaced pictures, each frame contains two picture fields:



The bits per frame can be measured by selecting the toolbar measurement tool, placing the crosshair cursor over the point to be measured, and reading the result from the left side of the status bar.

Audio trace enable

Audio Trace Enable

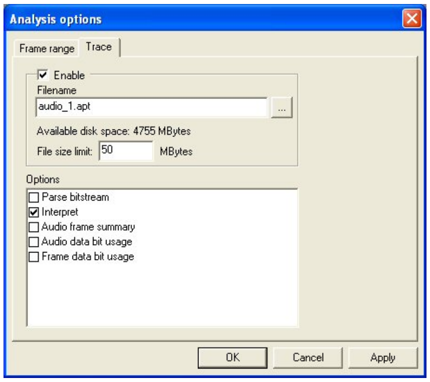


□ Enable
Filename: audio_3.apt
Available disk space: 4755 MBytes
File size limit: 50 MBytes

Options
- □ Parse bitstream
- □ Interpret
- □ Audio frame summary
- □ Main payload bit usage
- □ Subsidiary payload bit usage
- □ SBR bit usage
- □ Frame bit usage

The Trace options provide text outputs, which describe the contents of an encoded stream, to various levels of detail.

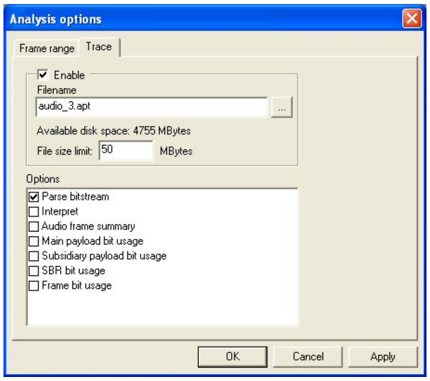
The available Trace options vary with each audio standard; see following sections (unless otherwise stated, each option applies to every audio standard).



☑ Enable
Filename: audio_1.apt
Available disk space: 4755 MBytes
File size limit: 50 MBytes

Options
- □ Parse bitstream
- ☑ Interpret
- □ Audio frame summary
- □ Audio data bit usage
- □ Frame data bit usage

MPEG-1/2 Audio



☑ Enable
Filename: audio_3.apt
Available disk space: 4755 MBytes
File size limit: 50 MBytes

Options
- ☑ Parse bitstream
- □ Interpret
- □ Audio frame summary
- □ Main payload bit usage
- □ Subsidiary payload bit usage
- □ SBR bit usage
- □ Frame bit usage

MPEG-2/4 AAC Audio

NOTE. *Using these options can generate a lot of data - files hundreds of megabytes or more in size. It may be advisable to collect the data on a range of frames, not all (this is a selectable option).*

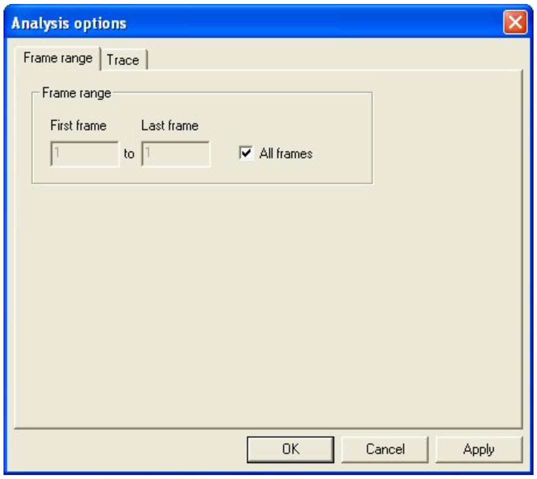
Alternatively, you can set a maximum file size, after which no more trace data is output to the selected file.

NOTE. *The audio can be stopped and the Trace file opened immediately after going past the last frame number in the frame range – there is no need to wait until the end of the sequence.*

NOTE. *To set the range of frames over which to collect the Trace data, click on the Frame range tab.*

Frame Range Tab

This option allows you to specify the start and end frames between which to gather the Trace information.



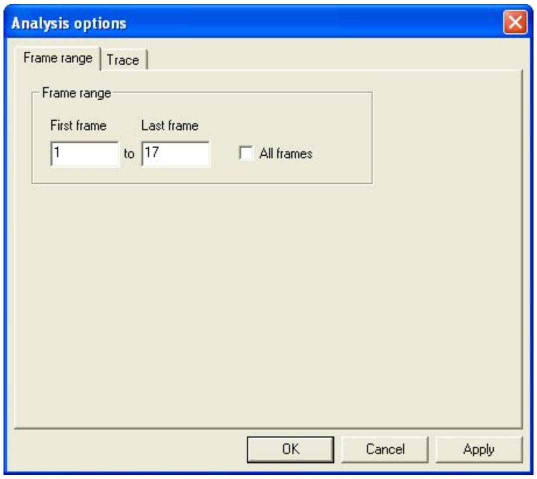
First frame 1 to Last frame 1

☑ All frames

NOTE. *Frame range... is unavailable in pause mode (indicated when the Pause icon ⏸ is pushed in on the toolbar and Play menu). This is because the frame range over which statistics are to be accumulated cannot be changed during pause mode. (If it could, you would not be sure that the Trace information was collected over the correct range of frames.) For example, if the audio was paused at frame 23 and Trace was enabled for a range of frames from 20-25, the Trace data would be incorrect.*

All frames. This option is the default and calculates the sequence statistics from the first to the last frame.

To set a range of frames. If the All frames check box is cleared, a range of frames can be entered (in this case, frames 1 to 17 inclusive):

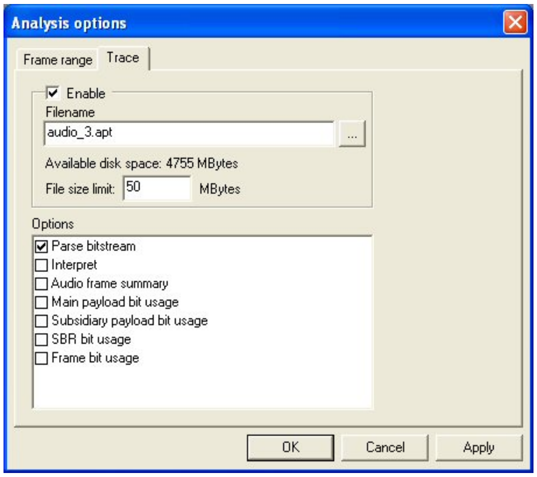


Frame range

First frame 1 to Last frame 17 □ All frames

Trace Tab

Enable. This option enables/disables the Trace output. If the output is disabled, the remaining selections on the Trace tab are unavailable, although the values are stored during the current use of MTS4EA (they are reset to their default values the next time the MTS4EA is started).



☑ Enable

Filename audio_3.apt

Available disk space: 4755 MBytes

File size limit: 50 MBytes

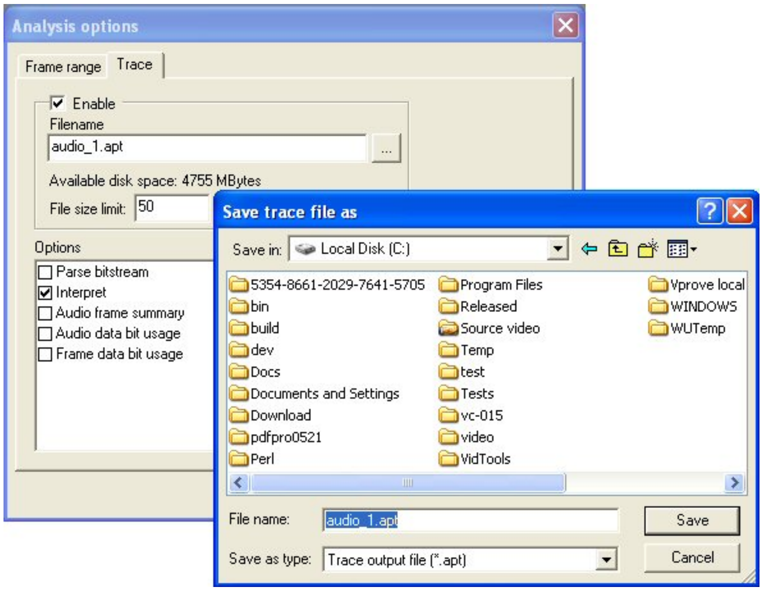
Options

- ☑ Parse bitstream
- □ Interpret
- □ Audio frame summary
- □ Main payload bit usage
- □ Subsidiary payload bit usage
- □ SBR bit usage
- □ Frame bit usage

Filename. By default, the filename suggested is the same as the base name of the input audio file, that is, the audio filename without the extension.

Any other filename can be entered; MTS4EA will append an `.apt` extension to this filename if it does not already end in this (`.apt` = MTS4EA trace).

To browse to a specific folder and enter the filename there, click on the [...] button:



Analysis options

Frame range | Trace

☑ Enable

Filename: audio_1.apt

Available disk space: 4755 MBytes

File size limit: 50

Options

□ Parse bitstream
☑ Interpret
□ Audio frame summary
□ Audio data bit usage
□ Frame data bit usage

Save trace file as

Save in: Local Disk (C:)

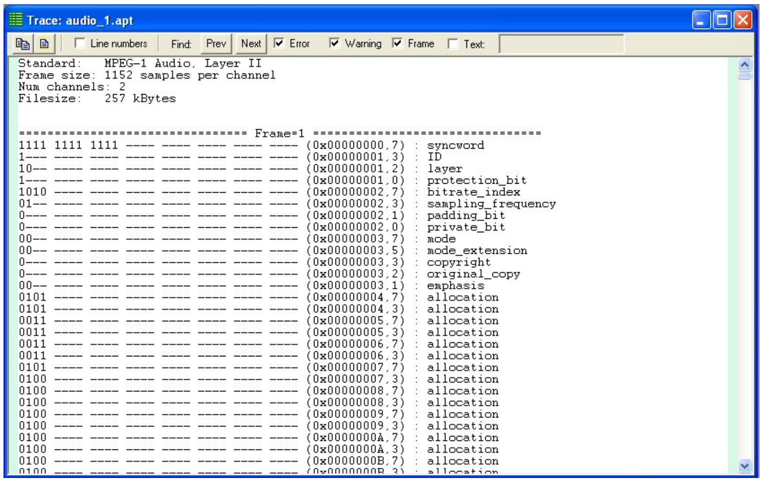
File name: audio_1.apt

Save as type: Trace output file (*.apt)

File size limit (available disk space). Trace files can generate a lot of data. This means that collecting data over a large number of frames can easily generate a Trace file hundreds of megabytes in size.

This entry allows you to limit the amount of disk space taken by the Trace file. Once the Trace file reaches this size, it stops writing more data.

Parse bitstream. If this option is selected, then the bitstream is parsed to a file. This gives the individual bit patterns for the bitstream fields as well as the mnemonic used within the standard to identify the field.



```
Standard:   MPEG-1 Audio, Layer II
Frame size: 1152 samples per channel
Num channels: 2
Filesize:   257 kBytes


================================ Frame=1 ================================
1111 1111 1111 ---- ---- ---- ---- ---- (0x00000000,7) : syncword
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000001,3) : ID
10-- ---- ---- ---- ---- ---- ---- ---- (0x00000001,2) : layer
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000001,0) : protection_bit
1010 ---- ---- ---- ---- ---- ---- ---- (0x00000002,7) : bitrate_index
01-- ---- ---- ---- ---- ---- ---- ---- (0x00000002,3) : sampling_frequency
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000002,1) : padding_bit
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000002,0) : private_bit
00-- ---- ---- ---- ---- ---- ---- ---- (0x00000003,7) : mode
00-- ---- ---- ---- ---- ---- ---- ---- (0x00000003,5) : mode_extension
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000003,3) : copyright
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000003,2) : original_copy
00-- ---- ---- ---- ---- ---- ---- ---- (0x00000003,1) : emphasis
0101 ---- ---- ---- ---- ---- ---- ---- (0x00000004,7) : allocation
0101 ---- ---- ---- ---- ---- ---- ---- (0x00000004,3) : allocation
0011 ---- ---- ---- ---- ---- ---- ---- (0x00000005,7) : allocation
0011 ---- ---- ---- ---- ---- ---- ---- (0x00000005,3) : allocation
0011 ---- ---- ---- ---- ---- ---- ---- (0x00000006,7) : allocation
0011 ---- ---- ---- ---- ---- ---- ---- (0x00000006,3) : allocation
0101 ---- ---- ---- ---- ---- ---- ---- (0x00000007,7) : allocation
0100 ---- ---- ---- ---- ---- ---- ---- (0x00000007,3) : allocation
0100 ---- ---- ---- ---- ---- ---- ---- (0x00000008,7) : allocation
0100 ---- ---- ---- ---- ---- ---- ---- (0x00000008,3) : allocation
0100 ---- ---- ---- ---- ---- ---- ---- (0x00000009,7) : allocation
0100 ---- ---- ---- ---- ---- ---- ---- (0x00000009,3) : allocation
0100 ---- ---- ---- ---- ---- ---- ---- (0x0000000A,7) : allocation
0100 ---- ---- ---- ---- ---- ---- ---- (0x0000000A,3) : allocation
0100 ---- ---- ---- ---- ---- ---- ---- (0x0000000B,7) : allocation
0100 ---- ---- ---- ---- ---- ---- ---- (0x0000000B,3) : allocation
```

The preceding example is the Parse bitstream output for an MPEG-1 Audio Layer II stream.

To quickly find if there are any errors in the Trace file, simply use the Find next button function and search on either ‘Error’ or ‘Warning’.

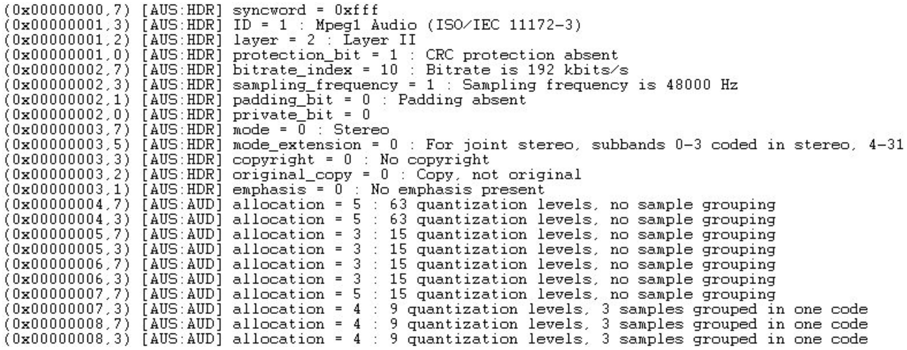
NOTE. *The data from the output of Parse bitstream is one of the best ways that MTS4EA provides to do bitstream syntax debugging.*

Go to other views/right click menu. A context-sensitive menu, brought up by a right-click of the mouse, is available in Interpret and Parse bitstream files. It allows you to:

- Copy selected trace data to the clipboard.
- Go to the area in another view that corresponds to the area that was selected in the Trace file.

NOTE. *The right-click ‘Goto view’ context-sensitive menu is only active in the Trace/Interpret and Trace/Parse Bitstream trace files.*

Interpret. This provides interpretation of the decoded data in the bitstream. Example outputs are shown below for MPEG-1:



```
(0x00000000,7) [AUS:HDR] syncword = 0xfff
(0x00000001,3) [AUS:HDR] ID = 1 : Mpeg1 Audio (ISO/IEC 11172-3)
(0x00000001,2) [AUS:HDR] layer = 2 : Layer II
(0x00000001,0) [AUS:HDR] protection_bit = 1 : CRC protection absent
(0x00000002,7) [AUS:HDR] bitrate_index = 10 : Bitrate is 192 kbits/s
(0x00000002,3) [AUS:HDR] sampling_frequency = 1 : Sampling frequency is 48000 Hz
(0x00000002,1) [AUS:HDR] padding_bit = 0 : Padding absent
(0x00000002,0) [AUS:HDR] private_bit = 0
(0x00000003,7) [AUS:HDR] mode = 0 : Stereo
(0x00000003,5) [AUS:HDR] mode_extension = 0 : For joint stereo, subbands 0-3 coded in stereo, 4-31
(0x00000003,3) [AUS:HDR] copyright = 0 : No copyright
(0x00000003,2) [AUS:HDR] original_copy = 0 : Copy, not original
(0x00000003,1) [AUS:HDR] emphasis = 0 : No emphasis present
(0x00000004,7) [AUS:AUD] allocation = 5 : 63 quantization levels, no sample grouping
(0x00000004,3) [AUS:AUD] allocation = 5 : 63 quantization levels, no sample grouping
(0x00000005,7) [AUS:AUD] allocation = 3 : 15 quantization levels, no sample grouping
(0x00000005,3) [AUS:AUD] allocation = 3 : 15 quantization levels, no sample grouping
(0x00000006,7) [AUS:AUD] allocation = 3 : 15 quantization levels, no sample grouping
(0x00000006,3) [AUS:AUD] allocation = 3 : 15 quantization levels, no sample grouping
(0x00000007,7) [AUS:AUD] allocation = 5 : 15 quantization levels, no sample grouping
(0x00000007,3) [AUS:AUD] allocation = 4 : 9 quantization levels, 3 samples grouped in one code
(0x00000008,7) [AUS:AUD] allocation = 4 : 9 quantization levels, 3 samples grouped in one code
(0x00000008,3) [AUS:AUD] allocation = 4 : 9 quantization levels, 3 samples grouped in one code
```

NOTE. *To quickly find out if there are any errors in the Trace file, simply use the Find next button function of View trace…, looking for 'Error' or 'Warning'.*

Go to other views/right-click menu. By right-clicking with the mouse, a context-sensitive menu appears (Interpret & Parse Bitstream only), which allows you to:

- Copy selected trace data to the clipboard.

- Go to the area in another view which corresponds to the area which has been selected in the Trace file.

Audio frame summary. This Trace output provides a series of data items dependent on the audio standard; for each individual audio frame, it will output data as listed here. All items after the first are per channel, and this portion of the list is repeated once for each channel in the audio stream. The n in square brackets is replaced by the index of the channel the data item refers to.

AAC:

- Frame number

- Total SBR data size[*n*] (bits)

- Total config data size[*n*] (bits)

- Total data size[*n*] (bits)

- Max scalefactor blocks[*n*] (blocks)

- Global gain[*n*]

- LTP data present[*n*]

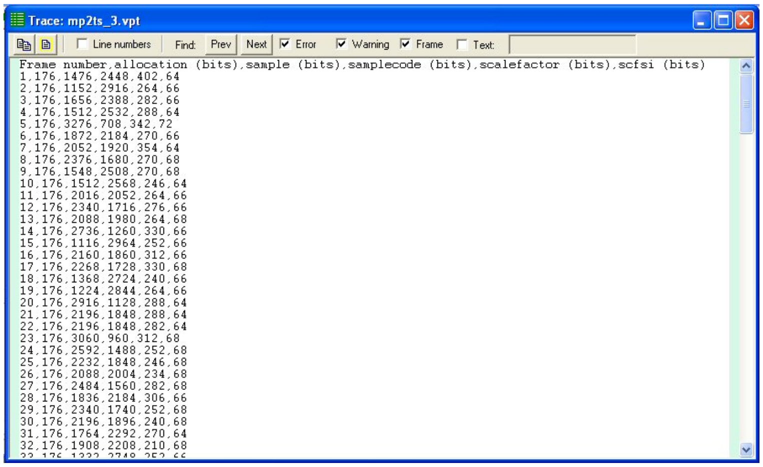
- Predictor data present[*n*]
- MS mask present[*n*]
- Pulse data present[*n*]
- TNS data present[*n*]
- Gain control data present[*n*]
- Window sequence[*n*]
- Window shape[*n*]

MPEG-2 Audio:

- Frame number
- Total audio data size[*n*] (bits)
- Ancillary data size[*n*] (bits)

The definition of each of these columns is given in the description of the Frame tooltip; see *Analysis Menu* section.

Audio data bit usage [MPEG-2].



```
Frame number,allocation (bits),sample (bits),samplecode (bits),scalefactor (bits),scfsi (bits)
1,176,1476,2448,402,64
2,176,1152,2916,264,66
3,176,1656,2388,282,66
4,176,1512,2532,288,64
5,176,3276,708,342,72
6,176,1872,2184,270,66
7,176,2052,1920,354,64
8,176,2376,1680,270,68
9,176,1548,2508,270,68
10,176,1512,2568,246,64
11,176,2016,2052,264,66
12,176,2340,1716,276,66
13,176,2088,1980,264,68
14,176,2736,1260,330,66
15,176,1116,2964,252,66
16,176,2160,1860,312,66
17,176,2268,1728,330,68
18,176,1368,2724,240,66
19,176,1224,2844,264,66
20,176,2916,1128,288,64
21,176,2196,1848,288,64
22,176,2196,1848,282,64
23,176,3060,960,312,68
24,176,2592,1488,252,68
25,176,2232,1848,246,68
26,176,2088,2004,234,68
27,176,2484,1560,282,68
28,176,1836,2184,306,66
29,176,2340,1740,252,68
30,176,2196,1896,240,68
31,176,1764,2292,270,64
32,176,1908,2208,210,68
33,176,1332,2748,252,66
```

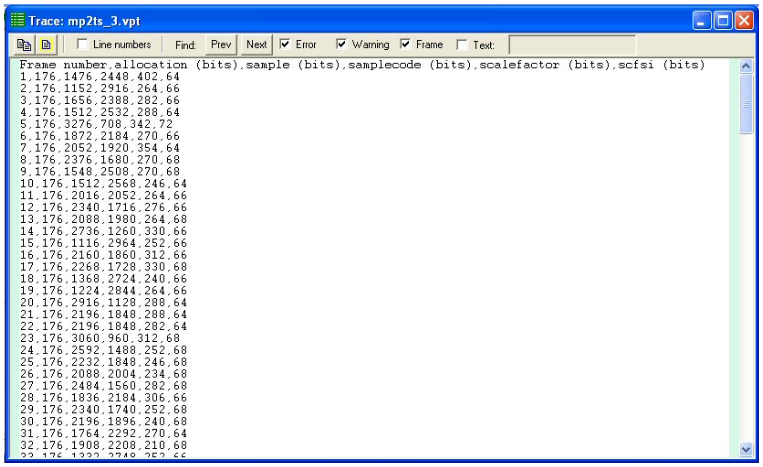
This Trace output provides a series of six data items, listed below, for each individual audio frame:

- Frame number

- allocation (bits)
- sample (bits)
- samplecode (bits)
- scalefactor (bits)
- scfsi (bits)

The definition of each of these columns is given in the description of the histograms; see *Show Histogram*, page 6–128.

Frame data bit usage [MPEG-2].



```
Frame number,allocation (bits),sample (bits),samplecode (bits),scalefactor (bits),scfsi (bits)
1,176,1476,2448,402,64
2,176,1152,2916,264,66
3,176,1656,2388,282,66
4,176,1512,2532,288,64
5,176,3276,708,342,72
6,176,1872,2184,270,66
7,176,2052,1920,354,64
8,176,2376,1680,270,68
9,176,1548,2508,270,68
10,176,1512,2568,246,64
11,176,2016,2052,264,66
12,176,2340,1716,276,66
13,176,2088,1980,264,68
14,176,2736,1260,330,66
15,176,1116,2964,252,66
16,176,2160,1860,312,66
17,176,2268,1728,330,68
18,176,1368,2724,240,66
19,176,1224,2844,264,66
20,176,2916,1128,288,64
21,176,2196,1848,288,64
22,176,2196,1848,282,64
23,176,3060,960,312,68
24,176,2592,1488,252,68
25,176,2232,1848,246,68
26,176,2088,2004,234,68
27,176,2484,1560,282,68
28,176,1836,2184,306,66
29,176,2340,1740,252,68
30,176,2196,1896,240,68
31,176,1764,2292,270,64
32,176,1908,2208,210,68
33,176,1332,2748,252,66
```

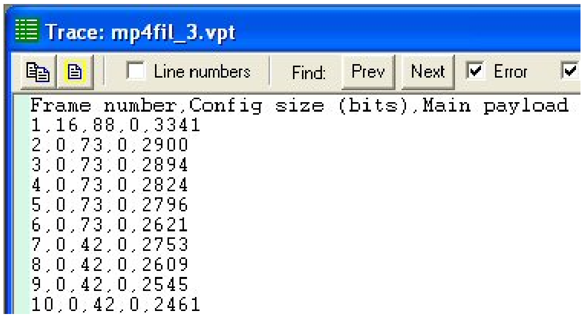
This Trace output provides a series of nine data items, listed below, for each individual audio frame:

- Frame number
- AncillaryData (bits)
- AudioData (bits)
- AudioSequence (bits)
- ErrorCheck (bits)
- Frame (bits)
- Header (bits)

- HuffmanCodeBits (bits)
- MainData (bits)

The definition of each of these columns is given in the description of the histograms; see Show Histogram, page 6–128.

Main payload bit usage [MPEG-4].



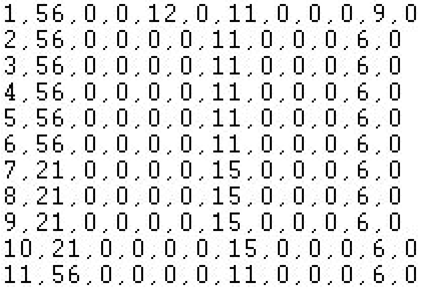
```
Frame number,Config size (bits),Main payload
1,16,88,0,3341
2,0,73,0,2900
3,0,73,0,2894
4,0,73,0,2824
5,0,73,0,2796
6,0,73,0,2621
7,0,42,0,2753
8,0,42,0,2609
9,0,42,0,2545
10,0,42,0,2461
```

This Trace output provides a series of 5 data items, listed below, for each individual audio frame:

- Frame number
- Config size (bits)
- Main payload size (bits)
- SBR size (bits)
- Subsidiary payload size (bits)

The definition of each of these columns is given in the description of the histograms; see Show Histogram, page 6–128.

Subsidiary payload bit usage [MPEG-4].



```
1,56,0,0,12,0,11,0,0,0,9,0
2,56,0,0,0,0,11,0,0,0,6,0
3,56,0,0,0,0,11,0,0,0,6,0
4,56,0,0,0,0,11,0,0,0,6,0
5,56,0,0,0,0,11,0,0,0,6,0
6,56,0,0,0,0,11,0,0,0,6,0
7,21,0,0,0,0,15,0,0,0,6,0
8,21,0,0,0,0,15,0,0,0,6,0
9,21,0,0,0,0,15,0,0,0,6,0
10,21,0,0,0,0,15,0,0,0,6,0
11,56,0,0,0,0,11,0,0,0,6,0
```

The following data items form the series of twelve seen in the Trace excerpt preceding (frames 1-11 shown):

- Frame number
- channel_pair_element (bits)
- coupling_channel_element (bits)
- data_stream_element (bits)
- fill_element (bits)
- gain_control_data (bits)
- ics_info (bits)
- lfe_channel_element (bits)
- program_config_element (bits)
- pulse_data (bits)
- raw_data_block (bits)
- single_channel_element (bits) (bits)

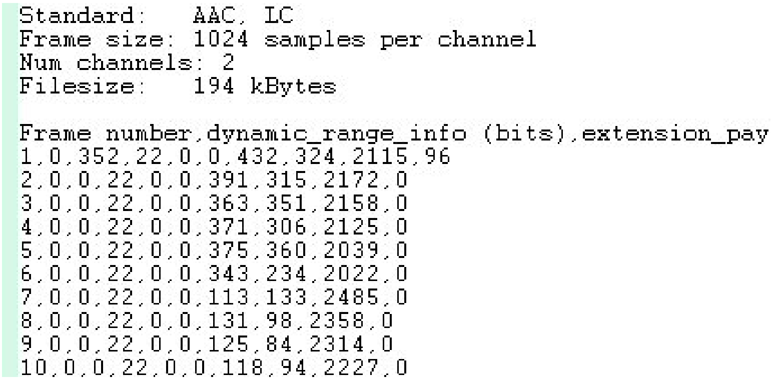
The definition of each of these columns is given in the description of the histograms; see Show Histogram, page 6–128.

SBR bit usage [MPEG-4]. Similar to the subsidiary payload bit usage Trace above, this Trace output provides a series of 16 data items, listed below, for each individual audio frame:

- Frame number
- sbr_channel_pair_base_element (bits)
- sbr_channel_pair_element (bits)
- sbr_channel_pair_enhance_element (bits)
- sbr_data (bits)
- sbr_dtdf (bits)
- sbr_envelope (bits)
- sbr_extension_data (bits)
- sbr_extension (bits)
- sbr_grid (bits)
- sbr_header (bits)
- sbr_invf (bits)
- sbr_noise (bits)
- sbr_single_channel_element (bits)
- sbr_sinusoidal_coding (bits)
- unused sbr data (no header found) (bits)

The definition of each of these columns is given in the description of the histograms; see Show Histogram, page 6–128.

Frame bit usage [MPEG-4].



```
Standard:   AAC, LC
Frame size: 1024 samples per channel
Num channels: 2
Filesize:   194 kBytes

Frame number,dynamic_range_info (bits),extension_pay
1,0,352,22,0,0,432,324,2115,96
2,0,0,22,0,0,391,315,2172,0
3,0,0,22,0,0,363,351,2158,0
4,0,0,22,0,0,371,306,2125,0
5,0,0,22,0,0,375,360,2039,0
6,0,0,22,0,0,343,234,2022,0
7,0,0,22,0,0,113,133,2485,0
8,0,0,22,0,0,131,98,2358,0
9,0,0,22,0,0,125,84,2314,0
10,0,0,22,0,0,118,94,2227,0
```

This Trace output provides a series of 10 data items, listed below, for each individual audio frame:

- Frame number
- dynamic_range_info (bits)
- extension_payload (bits)
- individual_channel_stream (bits)
- ltp_data (bits)
- reordered_spectral_data (bits)
- scale_factor_data (bits)
- section_data (bits)
- spectral_data (bits)
- tns_data (bits)

The definition of each of these columns is given in the description of the histograms; see Show Histogram, page 6–128.

Audio analysis options...

The Audio Analysis Options dialog box provides three tabs: Frame range, Trace and A/V delay. Frame range and Trace are described in *Audio trace*, page 6–203.

The A/V delay options are described in *Audio/Video Delay*, page 6–227.

View Audio trace... 🟩

This option opens the Trace view on the current output Trace file, as given in the Analysis menu Trace tab.

NOTE. *If there is not a current Trace file (and Trace is not enabled) this menu option/toolbar icon is unavailable (grayed out).*

NOTE. *Multiple View trace... windows can be opened at the same time.*

NOTE. *If the stream has been rerun and a Trace file by the same name is already open, then clicking the View trace... menu option or 🟩 icon will refresh the currently open Trace file.*

Go to other views/right click menu - Interpret & Parse bitstream only

By right-clicking, a context-sensitive menu appears, which allows you to go to the corresponding area in another view:

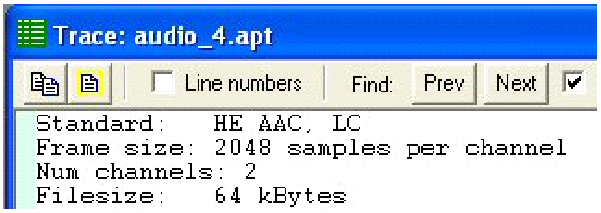
NOTE. *This right-click context-sensitive menu is only active in the Trace/Interpret and Trace/Parse Bitstream trace files.*

Copying Text

The contents of the Trace window can be copied to the Windows clipboard.

This can be done in a number of ways:

- By pressing <Ctrl+C>
- By right-clicking and selecting Copy
- By clicking the Copy icon (see below)

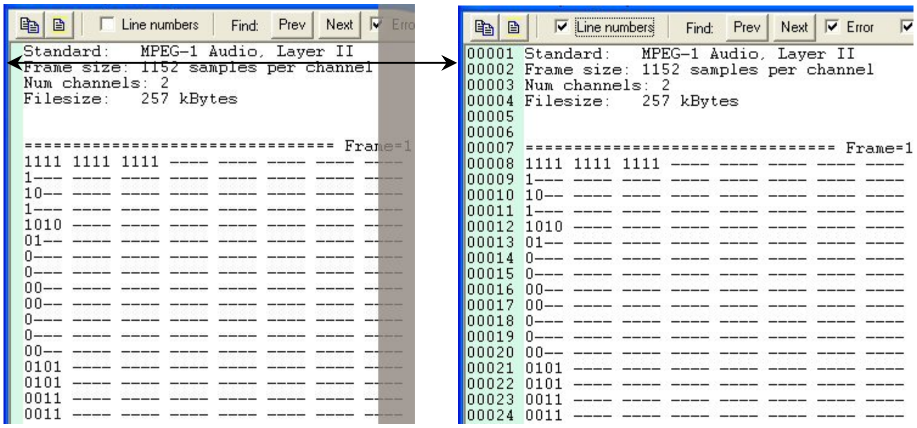


```
Standard:    HE AAC, LC
Frame size:  2048 samples per channel
Num channels: 2
Filesize:    64 kBytes
```

The icon to the immediate right of the Copy icon is the Select all icon. Rather than highlighting and dragging over what could be a lengthy Trace file, you can click on this icon to highlight all of the text in the Trace file for copying.

Line Numbers

Line numbers can be turned on/off:



```
Standard:    MPEG-1 Audio, Layer II
Frame size:  1152 samples per channel
Num channels: 2
Filesize:    257 kBytes


================================ Frame=1
1111 1111 1111 ---- ---- ---- ---- ----
1--- ---- ---- ---- ---- ---- ---- ----
10-- ---- ---- ---- ---- ---- ---- ----
1--- ---- ---- ---- ---- ---- ---- ----
1010 ---- ---- ---- ---- ---- ---- ----
01-- ---- ---- ---- ---- ---- ---- ----
0--- ---- ---- ---- ---- ---- ---- ----
0--- ---- ---- ---- ---- ---- ---- ----
00-- ---- ---- ---- ---- ---- ---- ----
00-- ---- ---- ---- ---- ---- ---- ----
0--- ---- ---- ---- ---- ---- ---- ----
0--- ---- ---- ---- ---- ---- ---- ----
00-- ---- ---- ---- ---- ---- ---- ----
0101 ---- ---- ---- ---- ---- ---- ----
0101 ---- ---- ---- ---- ---- ---- ----
0011 ---- ---- ---- ---- ---- ---- ----
0011 ---- ---- ---- ---- ---- ---- ----
```

```
00001 Standard:    MPEG-1 Audio, Layer II
00002 Frame size:  1152 samples per channel
00003 Num channels: 2
00004 Filesize:    257 kBytes
00005
00006
00007 ================================ Frame=1
00008 1111 1111 1111 ---- ---- ---- ---- ----
00009 1--- ---- ---- ---- ---- ---- ---- ----
00010 10-- ---- ---- ---- ---- ---- ---- ----
00011 1--- ---- ---- ---- ---- ---- ---- ----
00012 1010 ---- ---- ---- ---- ---- ---- ----
00013 01-- ---- ---- ---- ---- ---- ---- ----
00014 0--- ---- ---- ---- ---- ---- ---- ----
00015 0--- ---- ---- ---- ---- ---- ---- ----
00016 00-- ---- ---- ---- ---- ---- ---- ----
00017 00-- ---- ---- ---- ---- ---- ---- ----
00018 0--- ---- ---- ---- ---- ---- ---- ----
00019 0--- ---- ---- ---- ---- ---- ---- ----
00020 00-- ---- ---- ---- ---- ---- ---- ----
00021 0101 ---- ---- ---- ---- ---- ---- ----
00022 0101 ---- ---- ---- ---- ---- ---- ----
00023 0011 ---- ---- ---- ---- ---- ---- ----
00024 0011 ---- ---- ---- ---- ---- ---- ----
```

Find Data

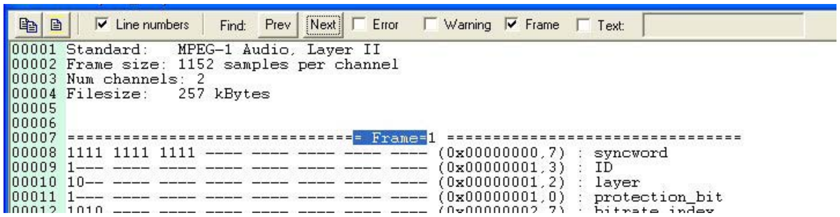
The Trace file can be searched for any data, using the Find: Prev (previous) and Next buttons. This finds the previous/next occurrence of any of the enabled strings.

NOTE. *The <Shift+F3> key can also be pressed, to find previous.*

There are some standard strings that are useful to search for:

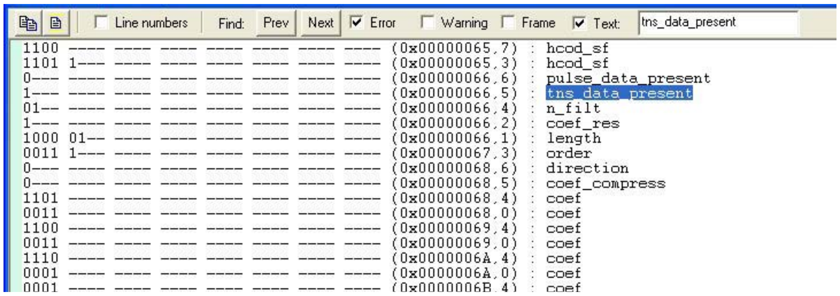
- Error
- Warning
- Frame

These are found by enabling the relevant check box (Frame example below):



```
00001 Standard:   MPEG-1 Audio, Layer II
00002 Frame size: 1152 samples per channel
00003 Num channels: 2
00004 Filesize:   257 kBytes
00005
00006
00007 ==============================  Frame=1 ===============================
00008 1111 1111 1111 ---- ---- ---- ---- ---- (0x00000000,7) : syncword
00009 1--- ---- ---- ---- ---- ---- ---- ---- (0x00000001,3) : ID
00010 10-- ---- ---- ---- ---- ---- ---- ---- (0x00000001,2) : layer
00011 1--- ---- ---- ---- ---- ---- ---- ---- (0x00000001,0) : protection_bit
00012 1010 ---- ---- ---- ---- ---- ---- ---- (0x00000002,7) : bitrate_index
```

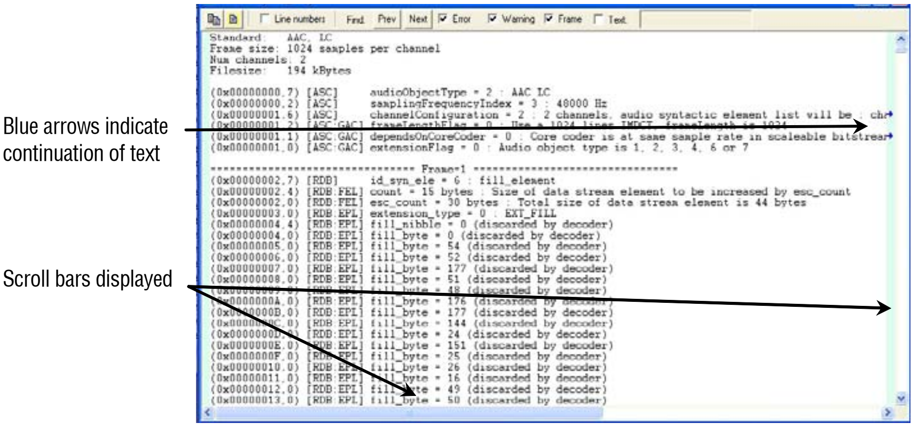
To search for any text, check the box next to Text and enter the text in the box (in this example, searching for 'tns_data_present'):



```
1100 ---- ---- ---- ---- ---- ---- ---- (0x00000065,7) : hcod_sf
1101 1--- ---- ---- ---- ---- ---- ---- (0x00000065,3) : hcod_sf
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000066,6) : pulse_data_present
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000066,5) : tns_data_present
01-- ---- ---- ---- ---- ---- ---- ---- (0x00000066,4) : n_filt
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000066,2) : coef_res
1000 01-- ---- ---- ---- ---- ---- ---- (0x00000066,1) : length
0011 1--- ---- ---- ---- ---- ---- ---- (0x00000067,3) : order
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000068,6) : direction
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000068,5) : coef_compress
1101 ---- ---- ---- ---- ---- ---- ---- (0x00000068,4) : coef
0011 ---- ---- ---- ---- ---- ---- ---- (0x00000068,0) : coef
1100 ---- ---- ---- ---- ---- ---- ---- (0x00000069,4) : coef
0011 ---- ---- ---- ---- ---- ---- ---- (0x00000069,0) : coef
1110 ---- ---- ---- ---- ---- ---- ---- (0x0000006A,4) : coef
0001 ---- ---- ---- ---- ---- ---- ---- (0x0000006A,0) : coef
0001 ---- ---- ---- ---- ---- ---- ---- (0x0000006B,4) : coef
```

Lines too long for Window Width

If a line is too long to be displayed within the width of the View trace window, a horizontal scroll bar is displayed and a blue arrow is appended to the right-hand side of the View trace window, to indicate the line extends to the right:

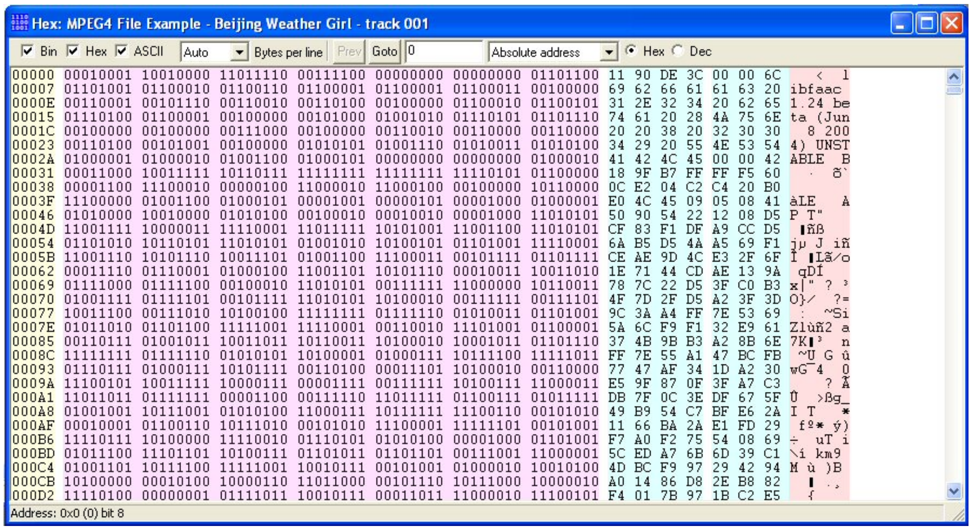


```
Standard:   AAC, LC
Frame size: 1024 samples per channel
Num channels: 2
Filesize:   194 kBytes

(0x00000000.7) [ASC]     audioObjectType = 2 : AAC LC
(0x00000000.2) [ASC]     samplingFrequencyIndex = 3 : 48000 Hz
(0x00000001.6) [ASC]     channelConfiguration = 2 : 2 channels, audio syntactic element list will be : ch
(0x00000001.2) [ASC:GAC] frameLengthFlag = 0 : Use a 1024 lines IMDCT, frameLength is 1024
(0x00000001.1) [ASC:GAC] dependsOnCoreCoder = 0 : Core coder is at same sample rate in scaleable bitstrea
(0x00000001.0) [ASC:GAC] extensionFlag = 0 : Audio object type is 1, 2, 3, 4, 6 or 7

******************************** Frame=1 ********************************
(0x00000002.7) [RDB]     id_syn_ele = 6 : fill_element
(0x00000002.4) [RDB:FEL] count = 15 bytes : Size of data stream element to be increased by esc_count
(0x00000002.0) [RDB:FEL] esc_count = 30 bytes : Total size of data stream element is 44 bytes
(0x00000003.0) [RDB:EPL] extension_type = 0 : EXT_FILL
(0x00000004.4) [RDB:EPL] fill_nibble = 0 (discarded by decoder)
(0x00000004.0) [RDB:EPL] fill_byte = 0 (discarded by decoder)
(0x00000005.0) [RDB:EPL] fill_byte = 54 (discarded by decoder)
(0x00000006.0) [RDB:EPL] fill_byte = 52 (discarded by decoder)
(0x00000007.0) [RDB:EPL] fill_byte = 177 (discarded by decoder)
(0x00000008.0) [RDB:EPL] fill_byte = 51 (discarded by decoder)
(0x00000009.0) [RDB:EPL] fill_byte = 48 (discarded by decoder)
(0x0000000A.0) [RDB:EPL] fill_byte = 176 (discarded by decoder)
(0x0000000B.0) [RDB:EPL] fill_byte = 177 (discarded by decoder)
(0x0000000C.0) [RDB:EPL] fill_byte = 144 (discarded by decoder)
(0x0000000D.0) [RDB:EPL] fill_byte = 24 (discarded by decoder)
(0x0000000E.0) [RDB:EPL] fill_byte = 151 (discarded by decoder)
(0x0000000F.0) [RDB:EPL] fill_byte = 25 (discarded by decoder)
(0x00000010.0) [RDB:EPL] fill_byte = 26 (discarded by decoder)
(0x00000011.0) [RDB:EPL] fill_byte = 16 (discarded by decoder)
(0x00000012.0) [RDB:EPL] fill_byte = 49 (discarded by decoder)
(0x00000013.0) [RDB:EPL] fill_byte = 50 (discarded by decoder)
```

Trace File Format

MTS4EA saves Trace files with a `.vpt` file extension. Currently, the.vpt file is a standard ASCII file which can be viewed in any text file viewer.

View audio stream hex...



Hex: MPEG4 File Example - Beijing Weather Girl - track 001

☑ Bin ☑ Hex ☑ ASCII Auto Bytes per line Prev Goto 0 Absolute address ● Hex ○ Dec

```
00000 00010001 10010000 11011110 00111100 00000000 00000000 01101100 11 90 DE 3C 00 00 6C    <  l
00007 01101001 01100010 01100110 01100001 01100001 01100011 00100000 69 62 66 61 61 63 20 ibfaac 
0000E 00110001 00101110 00110010 00110100 00100000 01100010 01100101 31 2E 32 34 20 62 65 1.24 be
00015 01110100 01100001 00100000 00101000 01001010 01110101 01101110 74 61 20 28 4A 75 6E ta (Jun
0001C 00100000 00100000 00111000 00100000 00110010 00110000 00110000 20 20 38 20 32 30 30   8 200
00023 00110100 00101001 00100000 01010101 01001110 01010011 01010100 34 29 20 55 4E 53 54 4) UNST
0002A 01000001 01000010 01001100 01000101 00000000 00000000 01000010 41 42 4C 45 00 00 42 ABLE  B
00031 00011000 10011111 10110111 11111111 11111111 11110101 01100000 18 9F B7 FF FF F5 60    ·  õ`
00038 00001100 11100010 00000100 11000010 11000100 00100000 10110000 0C E2 04 C2 C4 20 B0 
0003F 11100000 01001100 01000101 00001001 00000101 00001000 01000001 E0 4C 45 09 05 08 41 àLE    A
00046 01010000 10010000 01010100 00100010 00010010 00001000 11010101 50 90 54 22 12 08 D5 P T"
0004D 11001111 10000011 11110001 11011111 10101001 11001100 11010101 CF 83 F1 DF A9 CC D5  ▌ñß
00054 01101010 10110101 11010101 01001010 10100101 01101001 11110001 6A B5 D5 4A A5 69 F1 jµ J iñ
0005B 11001110 10101110 10011101 01001100 11100011 00101111 01101111 CE AE 9D 4C E3 2F 6F Î ▌Lã/o
00062 00011110 01110001 01000100 11001101 10101110 00010011 10011010 1E 71 44 CD AE 13 9A  qDÍ
00069 01111000 01111100 00100010 11010101 00111111 11000000 10110011 78 7C 22 D5 3F C0 B3 x|" ? ³
00070 01001111 01111101 00101111 11010101 10100010 00111111 00111101 4F 7D 2F D5 A2 3F 3D O}/  ?=
00077 10011100 00111010 10100100 11111111 01111110 01010011 01101001 9C 3A A4 FF 7E 53 69  :  ~Si
0007E 01011010 01101100 11111001 11110001 00110010 11101001 01100001 5A 6C F9 F1 32 E9 61 Zlùñ2 a
00085 00110111 01001011 10011011 10110011 10100010 10001011 01101110 37 4B 9B B3 A2 8B 6E 7K▌³  n
0008C 11111111 01111110 01010101 10100001 01000111 10111100 11111011 FF 7E 55 A1 47 BC FB  ~U G û
00093 01110111 01000111 10101111 00110100 00011101 10100010 00110000 77 47 AF 34 1D A2 30 wG¯4  0
0009A 11100101 10011111 10000111 00001111 00111111 10100111 11000011 E5 9F 87 0F 3F A7 C3     ? Ã
000A1 11011011 01111111 00001100 00111110 11011111 01100111 01011111 DB 7F 0C 3E DF 67 5F Û  >ßg_
000A8 01001001 10111001 01010100 11000111 10111111 11100110 00101010 49 B9 54 C7 BF E6 2A I T    *
000AF 00010001 01100110 10111010 00101010 11100001 11111101 00101001 11 66 BA 2A E1 FD 29  fº* ý)
000B6 11110111 10100000 11110010 01110101 01010100 00001000 01101001 F7 A0 F2 75 54 08 69 ÷  uT i
000BD 01011100 11101101 10100111 01101011 01101101 00111001 11000001 5C ED A7 6B 6D 39 C1 \í km9
000C4 01001101 10111100 11111001 10010111 00101001 01000010 10010100 4D BC F9 97 29 42 94 M ù )B
000CB 10100000 00010100 10000110 11011000 00101110 10111000 10000010 A0 14 86 D8 2E B8 82   ▌ .¸
000D2 11110100 00000001 01111011 10010111 00011011 11000010 11100101 F4 01 7B 97 1B C2 E5   {
```

Address: 0x0 (0) bit 8

This opens the current audio stream in a hex viewer, which shows the data in binary, hexadecimal, and ASCII data (in any combination of the three).

NOTE. *Multiple View hex... windows can be opened at the same time.*

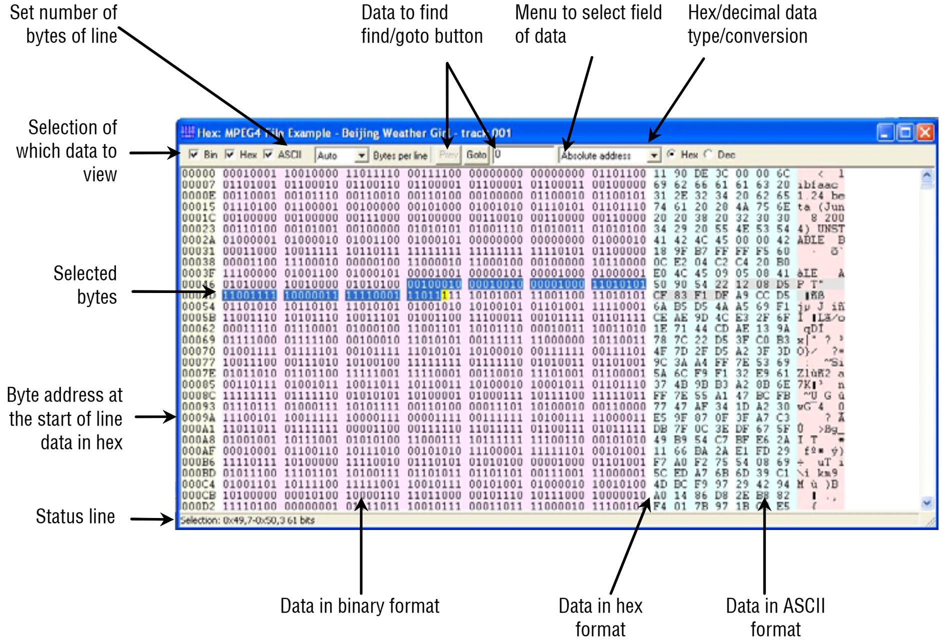
Using HexView, you can scroll through the file, and search for specific:

- Absolute address (from the start of the file)
- Relative address (from the currently-selected location)
- Binary data
- Hex data
- ASCII data

NOTE. *In the field where the data to find is entered, a wildcard character can be entered - this is '.' (a period).*

NOTE. *By entering a hex value and then clicking on Dec, you can convert the number to decimal (and vice versa).*

Sections of HexView window.



Set number of bytes of line
Data to find find/goto button
Menu to select field of data
Hex/decimal data type/conversion
Selection of which data to view
Selected bytes
Byte address at the start of line data in hex
Status line
Data in binary format
Data in hex format
Data in ASCII format

Hex: MPEG4 File Example - Beijing Weather Girl - track 001

☑ Bin ☑ Hex ☑ ASCII Auto Bytes per line Prev Goto 0 Absolute address ● Hex ○ Dec

```
00000 00010001 10010000 11011110 00111100 00000000 00000000 01101100 11 90 DE 3C 00 00 6C    <   l
00007 01101001 01100010 01100110 01100001 01100001 01100011 00100000 69 62 66 61 61 63 20 ibfaac
0000E 00110001 00101110 00110010 00110100 00100000 01100010 01100101 31 2E 32 34 20 62 65 1.24 be
00015 01110100 01100001 00100000 00101000 01001010 01110101 01101110 74 61 20 28 4A 75 6E ta (Jun
0001C 00100000 00100000 00111000 00100000 00110010 00110000 00110000 20 20 38 20 32 30 30   8 200
00023 00110100 00101001 00100000 01010101 01001110 01010011 01010100 34 29 20 55 4E 53 54 4) UNST
0002A 01000001 01000010 01001100 01000101 00000000 00000000 01000010 41 42 4C 45 00 00 42 ABLE  B
00031 00011000 10011111 10110111 11111111 11111111 11110101 01100000 18 9F B7 FF FF F5 60    ·  õ`
00038 00001100 11100010 00000100 11000010 11000100 00100000 10110000 0C E2 04 C2 C4 20 B0
0003F 11100000 01001100 01000101 00001001 00000101 00001000 01000001 E0 4C 45 09 05 08 41 àLE    A
00046 01010000 10010000 01010100 00100010 00010010 00001000 11010101 50 90 54 22 12 08 D5 P T"
0004D 11001111 10000011 11110001 11011111 10101001 11001100 11010101 CF 83 F1 DF A9 CC D5  ▮ñß
00054 01101010 10110101 11010101 01001010 10100101 01101001 11110001 6A B5 D5 4A A5 69 F1 jµ J iñ
0005B 11001110 10101110 10011101 01001100 11100011 00101111 01101111 CE AE 9D 4C E3 2F 6F Î ▮Lã/o
00062 00011110 01110001 01000100 11001101 10101110 00010011 10011010 1E 71 44 CD AE 13 9A  qDÍ
00069 01111000 01111100 00100010 11010101 00111111 11000000 10110011 78 7C 22 D5 3F C0 B3 x|" ? ³
00070 01001111 01111101 00101111 11010101 10100010 00111111 00111101 4F 7D 2F D5 A2 3F 3D O}/   ?=
00077 10011100 00111010 10100100 11111111 01111110 01010011 01101001 9C 3A A4 FF 7E 53 69  :   ~Si
0007E 01011010 01101100 11111001 11110001 00110010 11101001 01100001 5A 6C F9 F1 32 E9 61 Zlùñ2 a
00085 00110111 01001011 10011011 10110011 10100010 10001011 01101110 37 4B 9B B3 A2 8B 6E 7K▮³   n
0008C 11111111 01111110 01010101 10100001 01000111 10111100 11111011 FF 7E 55 A1 47 BC FB  ~U G û
00093 01110111 01000111 10101111 00110100 00011101 10100010 00110000 77 47 AF 34 1D A2 30 wG¯4   0
0009A 11100101 10011111 10000111 00001111 00111111 10100111 11000011 E5 9F 87 0F 3F A7 C3     ? Ã
000A1 11011011 01111111 00001100 00111110 11011111 01100111 01011111 DB 7F 0C 3E DF 67 5F Û   >Bg_
000A8 01001001 10111001 01010100 11000111 10111111 11100110 00101010 49 B9 54 C7 BF E6 2A I T    *
000AF 00010001 01100110 10111010 00101010 11100001 11111101 00101001 11 66 BA 2A E1 FD 29  fº* ý)
000B6 11110111 10100000 11110010 01110101 01010100 00001000 01101001 F7 A0 F2 75 54 08 69 ÷  uT i
000BD 01011100 11101101 10100111 01101011 01101101 00111001 11000001 5C ED A7 6B 6D 39 C1 \í km9
000C4 01001101 10111100 11111001 10010111 00101001 01000010 10010100 4D BC F9 97 29 42 94 M ù )B
000CB 10100000 00010100 10000110 11011000 00101110 10111000 10000010 A0 14 86 D8 2E B8 82   ▮ .,
000D2 11110100 00000001 01111011 10010111 00011011 11000010 11100101 F4 01 7B 97 1B C2 E5   {
```

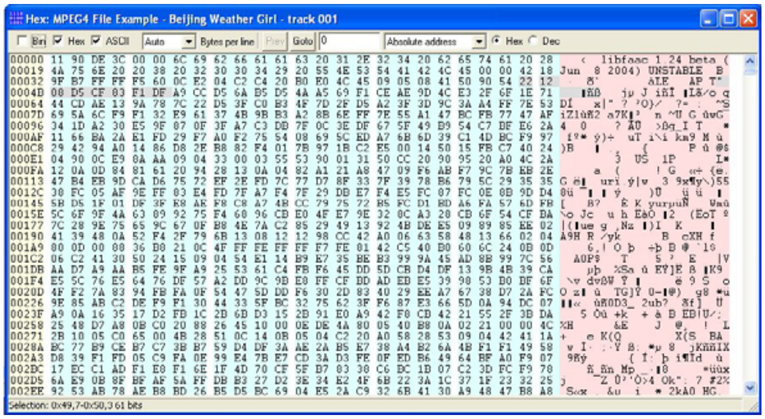
Selection: 0x49,7-0x50,3 61 bits

Setting information displayed; window width. The width and size of the HexView window can be set as with any other window in Windows. The same data is shown in each of the three sections:

- Binary
- Hexadecimal
- ASCII

Each of these areas can be individually displayed/hidden by clicking the appropriate tick box in the top left-hand corner of the window.

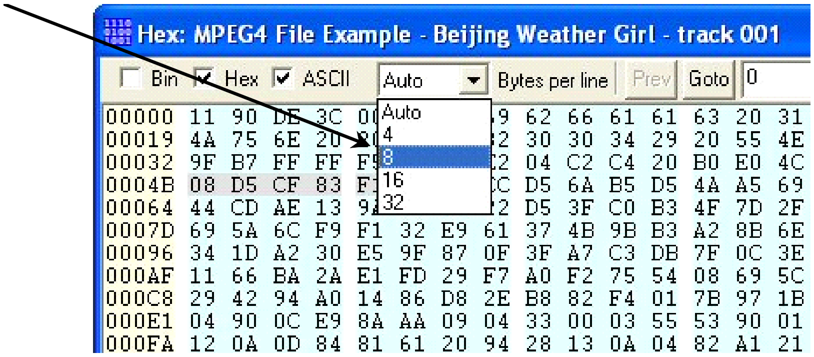
In this example the binary display has been turned off, leaving only hex and ASCII:



```
00000 11 90 DE 3C 00 00 6C 69 62 66 61 61 63 20 31 2E 32 34 20 62 65 74 61 20 28    <  libfaac 1.24 beta (
00019 4A 75 6E 20 20 38 20 32 30 30 34 29 20 55 4E 53 54 41 42 4C 45 00 00 42 18 Jun  8 2004) UNSTABLE  B
00032 9F B7 FF FF F5 60 0C E2 04 C2 C4 20 B0 E0 4C 45 09 05 08 41 50 90 54 22 12   .  ð'         àLE   AP T"
0004B 08 D5 CF 83 F1 DF A9 CC D5 6A B5 D5 4A A5 69 F1 CE AE 9D 4C E3 2F 6F 1E 71      ▮ñß    jµ J iñÎ ▮Lã/o q
00064 44 CD AE 13 9A 78 7C 22 D5 3F C0 B3 4F 7D 2F D5 A2 3F 3D 9C 3A A4 FF 7E 53 DÍ    x|" ? ³O}/  ?=  :  ~S
0007D 69 5A 6C F9 F1 32 E9 61 37 4B 9B B3 A2 8B 6E FF 7E 55 A1 47 BC FB 77 47 AF iZlùñ2 a7K▮³   n ~U G ûwG
00096 34 1D A2 30 E5 9F 87 0F 3F A7 C3 DB 7F 0C 3E DF 67 5F 49 B9 54 C7 DF E6 2A 4  0      ? ÃÛ   >ßg_I T     *
000AF 11 66 BA 2A E1 FD 29 F7 A0 F2 75 54 08 69 5C ED A7 6B 6D 39 C1 4D BC F9 97  f°* ý)÷  uT i\í km9 M ù
000C8 29 42 94 A0 14 86 D8 2E B8 82 F4 01 7B 97 1B C2 E5 00 14 50 15 FB C7 40 24 )B    ▮   .    {         P û @$
000E1 04 90 0C E9 0A AA 09 04 33 00 03 55 53 90 01 31 50 CC 20 90 95 20 A0 4C 2A             3  US  1P        L*
000FA 12 0A 0D 84 81 61 20 94 28 13 0A 04 82 A1 21 A8 47 09 F6 AB F7 9C 7B EB 2E          a  (        ! G  «÷ {ë.
00113 47 B4 EB 9D CA D6 75 72 EF 2E FD 7C 77 D7 8F 33 7F 39 78 B6 79 5C 29 35 35 G ë▮  urï.ý|w  3 9x¶y\)55
0012C 30 FC 05 AF 9E FF 03 E4 FD 7F A7 F4 7F 29 DB E7 F4 E5 FC 07 FC 0E 0B 9D D4 0ü   ▮ ▮ ý      )Û    ü ü   ▮
00145 5B D5 1F 01 DF 3F E8 AE F8 C8 A7 4B CC 79 75 72 B5 FC D1 BD A6 FA 57 6D FB [    ß?    È K yurµüÑ    Wmû
0015E 5C 6F 9F 4A 63 89 92 75 F4 60 96 CB E0 4F E7 9E 32 0C A3 20 CB 6F 54 CF DA \o Jc   u h ËàO ▮2   (ËoT º
00177 7C 28 9E 75 65 9C 67 0F B8 4E 7A C2 85 29 49 13 92 4B DE E5 09 89 85 EE 02 |(▮ue g ,Nz ▮)I   K        ▮
00190 41 39 48 0A 52 F4 2F 79 6B 13 08 12 12 98 CC 42 A0 06 63 58 48 13 66 02 04 A9H R /yk          B  cXH f
001A9 00 0D 00 00 36 D0 21 0C 4F FF FE FF FF F7 FE 01 42 C5 40 B0 60 6C 24 0B 0D       6,! O þ  ÷þ B @ `l$
001C2 06 C2 41 30 50 24 15 09 04 54 E1 14 B9 E7 35 BE B3 99 9A 45 AD 8B 99 7C 56    A0P$    T     5 ³   E    |V
001DB AA D7 A9 AA B5 FE 9F A9 25 53 61 C4 FB F6 45 DD 5D CB D4 DF 13 9B 4B 39 CA       µþ  %Sa û EÝ]Ë ß ▮K9
001F4 E5 5C 76 E5 64 76 DF 57 A2 DD 9C 9B E8 FF CF BD AD ED E5 39 98 53 B0 DF 6F   \v dvßW Ý ▮        ë 9 S   o
0020D 4F F2 7A 83 94 FB FA 0F 54 47 5D DD F6 30 2D 83 40 29 EE A7 67 38 D7 2A FC O z▮ û  TG]Ý 0-▮@)   g8 *ü
00226 9E 85 AB C2 DE F9 F1 30 44 33 5F BC 32 75 62 3F F6 87 E3 66 5D 0A 94 DC 07 ▮▮«  ùñ0D3_ 2ub?  ãf]   Ü
0023F A9 0A 16 35 17 D2 FB 1C 2B 6D D3 15 2B 91 E0 A9 42 F0 CD 42 21 55 2F 3B DA     5 Òû +k   + à B ÐB!U/;
00258 25 48 D7 A8 0B C0 20 88 26 45 10 00 0E DE 4A 80 05 40 B8 0A 02 21 00 00 4C %H          &E     J    @,    !   L
00271 2B 10 05 C0 65 00 4B 28 51 0C 14 0B 05 04 C2 20 A0 58 28 53 09 04 42 41 1A +    e K(Q          X(S  BA
0028A BC 77 B9 CE B7 C7 3B B7 59 D4 DF 3A AE 2A B5 E7 38 A4 B2 6A 4B F1 F1 49 58  w Î· ;·Y ß: *µ 8  jKññIX
002A3 D8 39 F1 FD 05 C9 FA 0E 99 E4 7B E7 CD 3A D3 FE 0F ED B6 49 64 BF A0 F9 07  9ñý        { Í: þ í¶Id  ù
002DC 17 EC C1 AD F1 E0 F1 6E 1F 4D 70 CF 5F D7 83 30 C6 DC 1D 07 C2 3D FC F9 78       ñ_ñn Mp _ ·▮0       =üùx
002D5 6A E9 0B 8F BF AF 5A FF DB B3 27 D2 3E 34 E2 4F 6B 22 3A 1C 37 1F 23 32 25 j        ¯Z Û³'Ò>4 âOk": 7 #2%
002EE 92 53 AB 78 AE B8 BD 26 B5 D5 BC 69 04 E5 2A C9 32 6B 41 30 A9 48 47 B8 A8  S«x .  &µ  i   * 2kA0 HG.
```

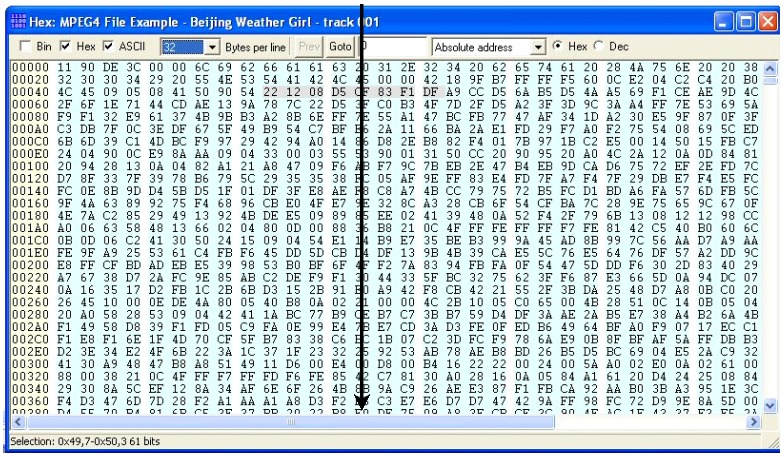
Selection: 0x49,7-0x50,3 61 bits

Setting bytes per line. This dropdown menu is used to set the number of bytes per line:



```
00000 11 90 DE 3C 00 ...9 62 66 61 61 63 20 31
00019 4A 75 6E 20 20 ...2 30 30 34 29 20 55 4E
00032 9F B7 FF FF F5 ...2 04 C2 C4 20 B0 E0 4C
0004B 08 D5 CF 83 F1 ...C D5 6A B5 D5 4A A5 69
00064 44 CD AE 13 9A ...2 D5 3F C0 B3 4F 7D 2F
0007D 69 5A 6C F9 F1 32 E9 61 37 4B 9B B3 A2 8B 6E
00096 34 1D A2 30 E5 9F 87 0F 3F A7 C3 DB 7F 0C 3E
000AF 11 66 BA 2A E1 FD 29 F7 A0 F2 75 54 08 69 5C
000C8 29 42 94 A0 14 86 D8 2E B8 82 F4 01 7B 97 1B
000E1 04 90 0C E9 8A AA 09 04 33 00 03 55 53 90 01
000FA 12 0A 0D 84 81 61 20 94 28 13 0A 04 82 A1 21
```

If the current width of the window is insufficient to show all the columns, then a scroll bar automatically appears at the bottom, allowing you to scroll left/right to see all the columns, as in the following example:



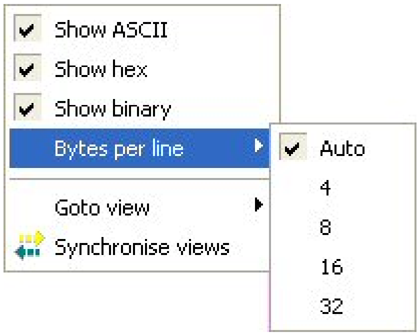
32

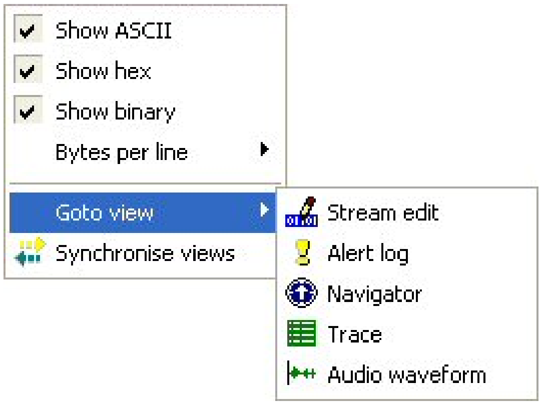
0

```
00000 11 90 DE 3C 00 00 6C 69 62 66 61 61 63 20 31 2E 32 34 20 62 65 74 61 20 28 4A 75 6E 20 20 38
00020 32 30 30 34 29 20 55 4E 53 54 41 42 4C 45 00 00 42 18 9F B7 FF FF F5 60 0C E2 04 C2 C4 20 B0
00040 4C 45 09 05 08 41 50 90 54 22 12 08 D5 CF 83 F1 DF A9 CC D5 6A B5 D5 4A A5 69 F1 CE AE 9D 4C
00060 2F 6F 1E 71 44 CD AE 13 9A 78 7C 22 D5 3F C0 B3 4F 7D 2F D5 A2 3F 3D 9C 3A A4 FF 7E 53 69 5A
00080 F9 F1 32 E9 61 37 4B 9B B3 A2 8B 6E FF 7E 55 A1 47 BC FB 77 47 AF 34 1D A2 30 E5 9F 87 0F 3F
000A0 C3 DB 7F 0C 3E DF 67 5F 49 B9 54 C7 BF E6 2A 11 66 BA 2A E1 FD 29 F7 A0 F2 75 54 08 69 5C ED
000C0 6B 6D 39 C1 4D BC F9 97 29 42 94 A0 14 86 D8 2E B8 82 F4 01 7B 97 1B C2 E5 00 14 50 15 FB C7
000E0 24 04 90 0C E9 8A AA 09 04 33 00 03 55 53 90 01 31 50 CC 20 90 95 20 A0 4C 2A 12 0A 0D 84 81
00100 20 94 28 13 0A 04 82 A1 21 A8 47 09 F6 AB F7 9C 7B EB 2E 47 B4 EB 9D CA D6 75 72 EF 2E FD 7C
00120 D7 8F 33 7F 39 78 B6 79 5C 29 35 35 38 FC 05 AF 9E FF 83 E4 FD 7F A7 F4 7F 29 DB E7 F4 E5 FC
00140 FC 0E 8B 9D D4 5B D5 1F 01 DF 3F E8 AE F8 C8 A7 4B CC 79 75 72 B5 FC D1 BD A6 FA 57 6D FB 5C
00160 9F 4A 63 89 92 75 F4 68 96 CB E0 4F E7 9E 32 8C A3 28 CB 6F 54 CF BA 7C 28 9E 75 65 9C 67 0F
00180 4E 7A C2 85 29 49 13 92 4B DE E5 09 89 85 EE 02 41 39 48 0A 52 F4 2F 79 6B 13 08 12 12 98 CC
001A0 A0 06 63 58 48 13 66 02 04 80 0D 00 88 36 B8 21 0C 4F FF FE FF FF F7 FE 81 42 C5 40 B0 60 6C
001C0 0B 0D 06 C2 41 30 50 24 15 09 04 54 E1 14 B9 E7 35 BE B3 99 9A 45 AD 8B 99 7C 56 AA D7 A9 AA
001E0 FE 9F A9 25 53 61 C4 FB F6 45 DD 5D CB D4 DF 13 9B 4B 39 CA E5 5C 76 E5 64 76 DF 57 A2 DD 9C
00200 E8 FF CF BD AD EB E5 39 98 53 B0 BF 6F 4F F2 7A 83 94 FB FA 0F 54 47 5D DD F6 30 2D 83 40 29
00220 A7 67 38 D7 2A FC 9E 85 AB C2 DE F9 F1 30 44 33 5F BC 32 75 62 3F F6 87 E3 66 5D 0A 94 DC 07
00240 0A 16 35 17 D2 FB 1C 2B 6B D3 15 2B 91 E0 A9 42 F8 CB 42 21 55 2F 3B DA 25 48 D7 A8 0B C0 20
00260 26 45 10 00 0E DE 4A 80 05 40 B8 0A 02 21 00 00 4C 2B 10 05 C0 65 00 4B 28 51 0C 14 0B 05 04
00280 20 A0 58 28 53 09 04 42 41 1A BC 77 B9 CE B7 C7 3B B7 59 D4 DF 3A AE 2A B5 E7 38 A4 B2 6A 4B
002A0 F1 49 58 D8 39 F1 FD 05 C9 FA 0E 99 E4 7B E7 CD 3A D3 FE 0F ED B6 49 64 BF A0 F9 07 17 EC C1
002C0 F1 E8 F1 6E 1F 4D 70 CF 5F B7 83 38 C6 EC 1B 07 C2 3D FC F9 78 6A E9 0B 8F BF AF 5A FF DB B3
002E0 D2 3E 34 E2 4F 6B 22 3A 1C 37 1F 23 32 25 92 53 AB 78 AE B8 BD 26 B5 D5 BC 69 04 E5 2A C9 32
00300 41 30 A9 48 47 B8 A8 51 49 11 D6 00 E4 00 D8 00 B4 16 22 22 00 24 00 5A A0 02 E0 0A 02 61 00
00320 88 00 38 21 0C 4F FF F7 FF FD F6 FE 85 42 C7 81 30 A0 28 16 0A 05 84 A1 61 20 D4 24 25 08 84
00340 29 30 8A 5C EF 12 8A 34 AF 6E 6F 26 4B 8B 9A C9 26 AE E3 87 F1 FB CA 92 AA B0 3B A3 95 1E 3C
00360 F4 D3 47 6D 7D 28 F2 A1 AA A1 A8 D3 F2 E5 C3 E7 E6 D7 D7 47 42 9A FF 98 FC 72 D9 9E 8A 5D 00
```

Selection: 0x49,7-0x50,3 61 bits

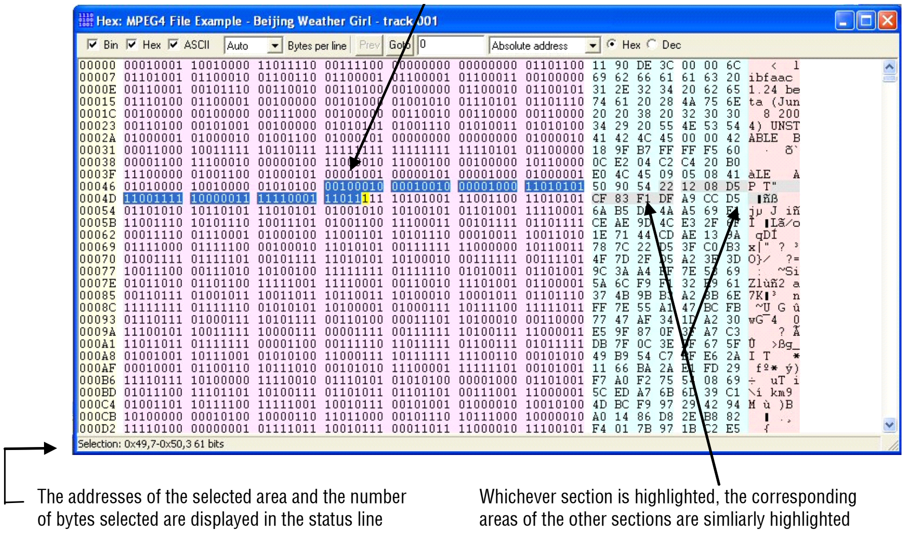
Right-click popup menu/Goto view. Right-clicking in the HexView window generates a menu, from which it is possible both to set the number of bytes displayed per line and to navigate to the same location within the stream in an alternative view:





Synchronize views: When the Synchronize views icon is pushed in, all open windows automatically follow the selection that you make.

Highlighting a section. A section of data can be highlighted in any of the (visible) binary, hex or ASCII sections by clicking and dragging the mouse over the area:



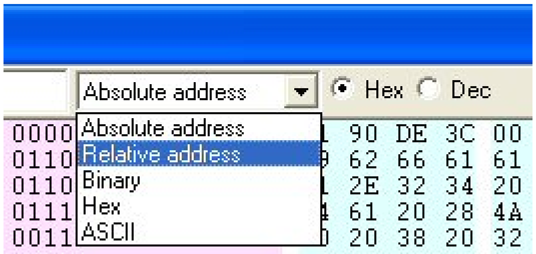
```
00000 00010001 10010000 11011110 00111100 00000000 00000000 01101100 11 90 DE 3C 00 00 6C    <   l
00007 01101001 01100010 01100110 01100001 01100001 01100011 00100000 69 62 66 61 61 63 20 ibfaac
0000E 00110001 00101110 00110010 00110100 00100000 01100010 01100101 31 2E 32 34 20 62 65 1.24 be
00015 01110100 01100001 00100000 00101000 01001010 01110101 01101110 74 61 20 28 4A 75 6E ta (Jun
0001C 00100000 00100000 00111000 00100000 00110010 00110000 00110000 20 20 38 20 32 30 30   8 200
00023 00110100 00101001 00100000 01010101 01001110 01010011 01010100 34 29 20 55 4E 53 54 4) UNST
0002A 01000001 01000010 01001100 01000101 00000000 00000000 01000010 41 42 4C 45 00 00 42 ABLE  B
00031 00011000 10011111 10110111 11111111 11111111 11110101 01100000 18 9F B7 FF FF F5 60   ·  õ`
00038 00001100 11100010 00000100 11000010 11000100 00100000 10110000 0C E2 04 C2 C4 20 B0
0003F 11100000 01001100 01000101 00001001 00000101 00001000 01000001 E0 4C 45 09 05 08 41 àLE   A
00046 01010000 10010000 01010100 00100010 00010010 00001000 11010101 50 90 54 22 12 08 D5 P T"
0004D 11001111 10000011 11110001 11011111 10101001 11001100 11010101 CF 83 F1 DF A9 CC D5  ▮ñß
00054 01101010 10110101 11010101 01001010 10100101 01101001 11110001 6A B5 D5 4A A5 69 F1 jµ J iñ
0005B 11001110 10101110 10011101 01001100 11100011 00101111 01101111 CE AE 9D 4C E3 2F 6F Î ▮Lã/o
00062 00011110 01110001 01000100 11001101 10101110 00010011 10011010 1E 71 44 CD AE 13 9A  qDÍ
00069 01111000 01111100 00100010 11010101 00111111 11000000 10110011 78 7C 22 D5 3F C0 B3 x|" ? ³
00070 01001111 01111101 00101111 11010101 10100010 00111111 00111101 4F 7D 2F D5 A2 3F 3D O}/  ?=
00077 10011100 00111010 10100100 11111111 01111110 01010011 01101001 9C 3A A4 FF 7E 53 69  :  ~Si
0007E 01011010 01101100 11111001 11110001 00110010 11101001 01100001 5A 6C F9 F1 32 E9 61 Zlùñ2 a
00085 00110111 01001011 10011011 10110011 10100010 10001011 01101110 37 4B 9B B3 A2 8B 6E 7K▮³  n
0008C 11111111 01111110 01010101 10100001 01000111 10111100 11111011 FF 7E 55 A1 47 BC FB  ~U G û
00093 01110111 01000111 10101111 00110100 00011101 10100010 00110000 77 47 AF 34 1D A2 30 wG¯4  0
0009A 11100101 10011111 10000111 00001111 00111111 10100111 11000011 E5 9F 87 0F 3F A7 C3     ? Ã
000A1 11011011 01111111 00001100 00111110 11011111 01100111 01011111 DB 7F 0C 3E DF 67 5F Û  >ßg_
000A8 01001001 10111001 01010100 11000111 10111111 11100110 00101010 49 B9 54 C7 BF E6 2A I T    *
000AF 00010001 01100110 10111010 00101010 11100001 11111101 00101001 11 66 BA 2A E1 FD 29  fº* ý)
000B6 11110111 10100000 11110010 01110101 01010100 00001000 01101001 F7 A0 F2 75 54 08 69 ÷  uT i
000BD 01011100 11101101 10100111 01101011 01101101 00111001 11000001 5C ED A7 6B 6D 39 C1 \í km9
000C4 01001101 10111100 11111001 10010111 00101001 01000010 10010100 4D BC F9 97 29 42 94 M ù )B
000CB 10100000 00010100 10000110 11011000 00101110 10111000 10000010 A0 14 86 D8 2E B8 82  ▮ .‚
000D2 11110100 00000001 01111011 10010111 00011011 11000010 11100101 F4 01 7B 97 1B C2 E5  {
Selection: 0x49,7-0x50,3 61 bits
```

The addresses of the selected area and the number of bytes selected are displayed in the status line

Whichever section is highlighted, the corresponding areas of the other sections are simliarly highlighted

Find absolute address. A specific address can be found in the audio stream, either:

- An absolute address, from the start of the audio file
- A relative address, from the first byte of the currently selected area



The address to find is entered in the box next to the Goto button. This address can be in hex or decimal.

If an invalid character is entered for an address (entering anything other than '0-9' or 'a-f' for a hexadecimal address), the Goto button is greyed out.

Note that the Goto button changes to Next for all data to find except for the Absolute address. (When Absolute address is selected, the Prev button is greyed out.)

The Hex/Dec option button selects the format of the data being searched - hexademical or decimal.

Find relative address. An address relative to the currently highlighted address can be found. If there is no address highlighted, the address found is the offset from 0.

The address to find is entered in the box next to the Next button. This address can be in hex or decimal. If an invalid character is entered for an address (for example, entering anything other than '0-9' or 'a-f' for a hexadecimal address), the Find button is greyed out.

NOTE. *The <F3> key can also be pressed to find next. The <Shift+F3> key can also be pressed to find previous.*

The Hex/Dec radio button selects the format of the data being searched.

Find Binary/Hex/ASCII. These menu options find data in the bitstream. Up to 64 characters can be entered.

NOTE. *The Binary search searches for the bit pattern irrespective of byte location; the Hex and ASCII searches are byte aligned.*

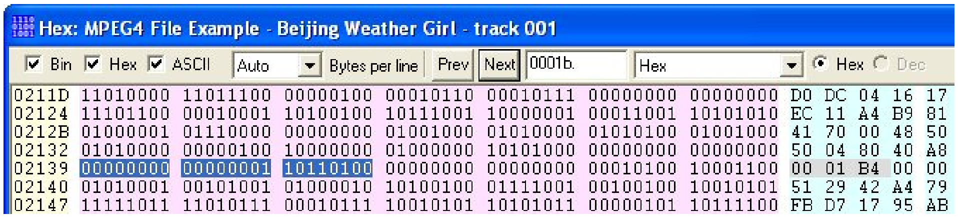
NOTE. *The <F3> key can also be pressed to find next. The <Shift+F3> key can also be pressed to find previous.*

Wildcard searching using '.'. A wildcard can be entered in the data to be found, this is the period character - '.'

The wildcard matches a single digit in the base selected; the wildcard matches:

- One bit in the binary find
- A hex digit in the hex find
- An ASCII character in the ASCII find

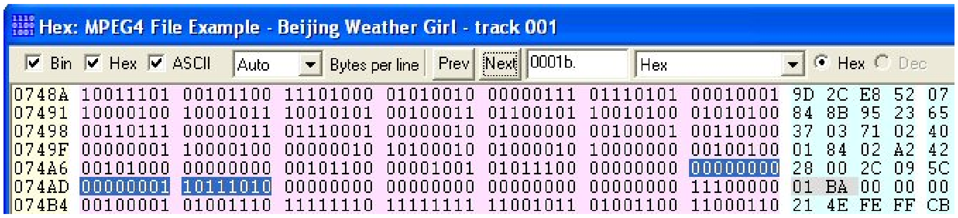
For example, entering '0001b.' as a hex string to find (any character after the '0001b' pattern will find the following patterns in the selected example stream:



Hex: MPEG4 File Example - Beijing Weather Girl - track 001

☑ Bin ☑ Hex ☑ ASCII | Auto | Bytes per line | Prev | Next | 0001b. | Hex | ● Hex ○ Dec

```
0211D 11010000 11011100 00000100 00010110 00010111 00000000 00000000 D0 DC 04 16 17
02124 11101100 00010001 10100100 10111001 10000001 00011001 10101010 EC 11 A4 B9 81
0212B 01000001 01110000 00000000 01001000 01010000 01010100 01001000 41 70 00 48 50
02132 01010000 00000100 10000000 01000000 10101000 00000000 00000000 50 04 80 40 A8
02139 00000000 00000001 10110100 00000000 00000000 00010100 10001100 00 01 B4 00 00
02140 01010001 00101001 01000010 10100100 01111001 00100100 10010101 51 29 42 A4 79
02147 11111011 11010111 00010111 10010101 10101011 00000101 10111100 FB D7 17 95 AB
```



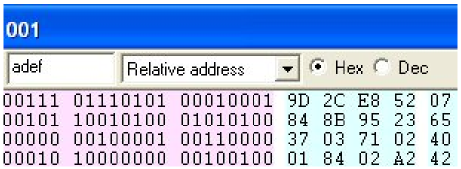
Hex: MPEG4 File Example - Beijing Weather Girl - track 001

☑ Bin ☑ Hex ☑ ASCII | Auto | Bytes per line | Prev | Next | 0001b. | Hex | ● Hex ○ Dec

```
0748A 10011101 00101100 11101000 01010010 00000111 01110101 00010001 9D 2C E8 52 07
07491 10000100 10001011 10010101 00100011 01100101 10010100 01010100 84 8B 95 23 65
07498 00110111 00000011 01110001 00000010 01000000 00100001 00110000 37 03 71 02 40
0749F 00000001 10000100 00000010 10100010 01000010 10000000 00100100 01 84 02 A2 42
074A6 00101000 00000000 00101100 00001001 01011100 00000000 00000000 28 00 2C 09 5C
074AD 00000001 10111010 00000000 00000000 00000000 00000000 11100000 01 BA 00 00 00
074B4 00100001 01001110 11111110 11111111 11001011 01001100 11000110 21 4E FE FF CB
```

The wildcard can be inserted in any position. For example, '03..456.8' is a valid search string, which will find any nine consecutive digits where the first two are '03', digits 5-7 are '456' and the last digit is '8'.

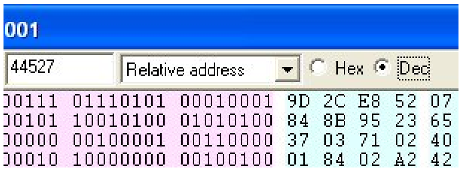
Conversion of hex<->decimal. If a hex value is entered into the find box, and the Dec option button is clicked, the value is converted to hex (and vice versa):



001

adef | Relative address | ● Hex ○ Dec

```
00111 01110101 00010001 9D 2C E8 52 07
00101 10010100 01010100 84 8B 95 23 65
00000 00100001 00110000 37 03 71 02 40
00010 10000000 00100100 01 84 02 A2 42
```



001

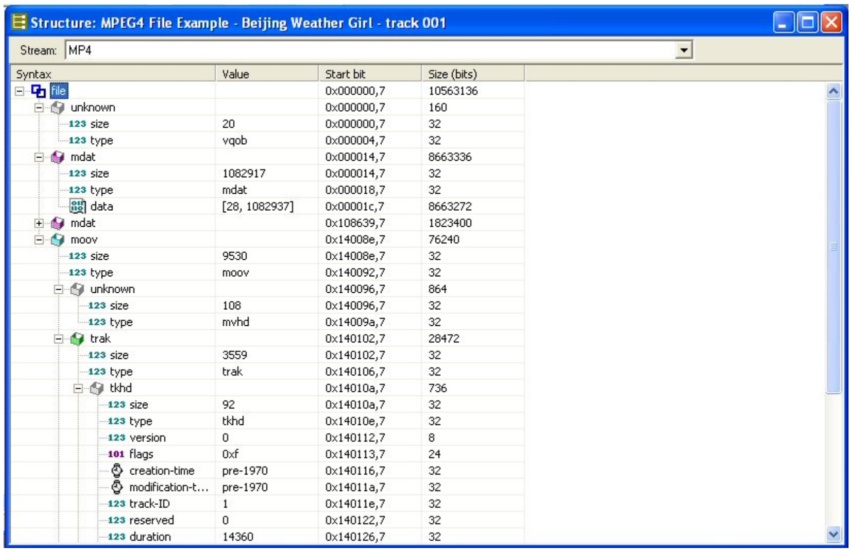
44527 | Relative address | ○ Hex ● Dec

```
00111 01110101 00010001 9D 2C E8 52 07
00101 10010100 01010100 84 8B 95 23 65
00000 00100001 00110000 37 03 71 02 40
00010 10000000 00100100 01 84 02 A2 42
```

View audio stream structure...

This allows the structure of the following file types to be viewed and expanded/contracted (by clicking on the '-' and '+' symbols at the left end of each line). Each branch of the structure can be clicked on to expand it and see the nodes below it - see the examples following:

- MP4 containing AAC LC:

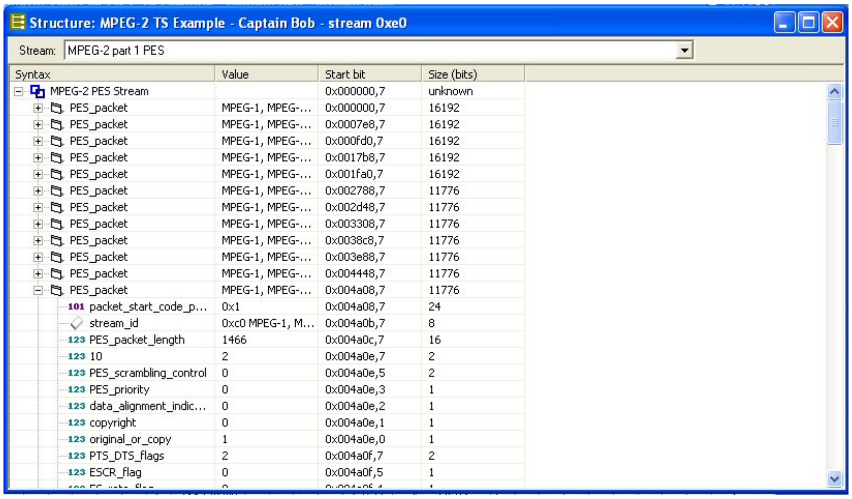


Structure: MPEG4 File Example - Beijing Weather Girl - track 001

Stream: MP4

Syntax	Value	Start bit	Size (bits)
file		0x000000,7	10563136
unknown		0x000000,7	160
size	20	0x000000,7	32
type	vqob	0x000004,7	32
mdat		0x000014,7	8663336
size	1082917	0x000014,7	32
type	mdat	0x000018,7	32
data	[28, 1082937]	0x00001c,7	8663272
mdat		0x108639,7	1823400
moov		0x14008e,7	76240
size	9530	0x14008e,7	32
type	moov	0x140092,7	32
unknown		0x140096,7	864
size	108	0x140096,7	32
type	mvhd	0x14009a,7	32
trak		0x140102,7	28472
size	3559	0x140102,7	32
type	trak	0x140106,7	32
tkhd		0x14010a,7	736
size	92	0x14010a,7	32
type	tkhd	0x14010e,7	32
version	0	0x140112,7	8
flags	0xf	0x140113,7	24
creation-time	pre-1970	0x140116,7	32
modification-t...	pre-1970	0x14011a,7	32
track-ID	1	0x14011e,7	32
reserved	0	0x140122,7	32
duration	14360	0x140126,7	32

- Container files, for example, MPEG-2 PES:



Structure: MPEG-2 TS Example - Captain Bob - stream 0xe0

Stream: MPEG-2 part 1 PES

Syntax	Value	Start bit	Size (bits)
MPEG-2 PES Stream		0x000000,7	unknown
PES_packet	MPEG-1, MPEG-...	0x000000,7	16192
PES_packet	MPEG-1, MPEG-...	0x0007e8,7	16192
PES_packet	MPEG-1, MPEG-...	0x000fd0,7	16192
PES_packet	MPEG-1, MPEG-...	0x0017b8,7	16192
PES_packet	MPEG-1, MPEG-...	0x001fa0,7	16192
PES_packet	MPEG-1, MPEG-...	0x002788,7	11776
PES_packet	MPEG-1, MPEG-...	0x002d48,7	11776
PES_packet	MPEG-1, MPEG-...	0x003308,7	11776
PES_packet	MPEG-1, MPEG-...	0x0038c8,7	11776
PES_packet	MPEG-1, MPEG-...	0x003e88,7	11776
PES_packet	MPEG-1, MPEG-...	0x004448,7	11776
PES_packet	MPEG-1, MPEG-...	0x004a08,7	11776
packet_start_code_p...	0x1	0x004a08,7	24
stream_id	0xc0 MPEG-1, M...	0x004a0b,7	8
PES_packet_length	1466	0x004a0c,7	16
10	2	0x004a0e,7	2
PES_scrambling_control	0	0x004a0e,5	2
PES_priority	0	0x004a0e,3	1
data_alignment_indic...	0	0x004a0e,2	1
copyright	0	0x004a0e,1	1
original_or_copy	1	0x004a0e,0	1
PTS_DTS_flags	2	0x004a0f,7	2
ESCR_flag	0	0x004a0f,5	1

The view is divided into four columns:

Syntax. This column shows the stream syntax parse tree, including the names of each syntax element or syntactic structure. Nested syntax structures are shown by nesting nodes of the tree. An icon shows the type of data at each tree node.

Value. This column shows the interpreted value of syntax elements in the tree. The content of this field will vary according to the type of data to be shown.

Start bit. Shows the start bit address of the syntax node. The address is given as a hexadecimal byte position followed by a bit position number.

Size. Shows the size in bits of the syntax node. For nodes with children, this shows the total bits of the node and all its children.

See also *View Video Stream Structure...* ▤ *Ctrl+R*, page 6–184.

Audio/Video Delay

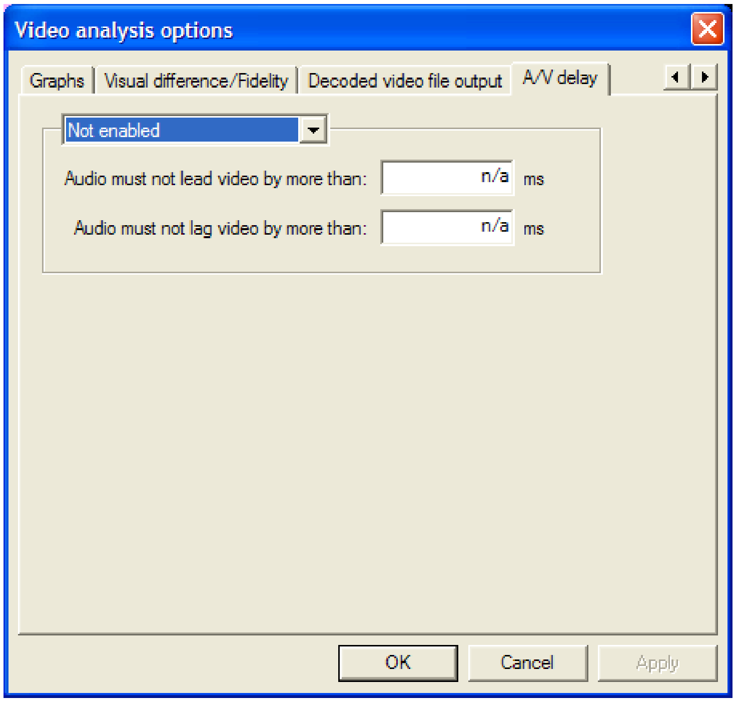
This feature provides a complete solution for measuring audio/video synchronization delay in an encoding system. MTS4EA provides a selection of predefined, clock-tick test streams in various formats. You can select an audio and a video stream and encode them using your own equipment. You can then analyze the resulting encoded stream using the MTS4EA and measure any time delay between the audio and video that may have been introduced by the encoding.

The available test streams include uncompressed streams for testing elementary stream encoders and compressed streams for testing transport multiplexers.

(See also *Export A/V Delay Streams…*, page 6–44.)

A/V Delay Test Enable…

This menu option opens the Video Analysis Options dialog box. This option, when used with the supplied A/V Delay streams, allows you to measure audio/video delay, which may have been introduced when encoding elementary streams. When enabled, the feature allows you to select preset or user selected delays.

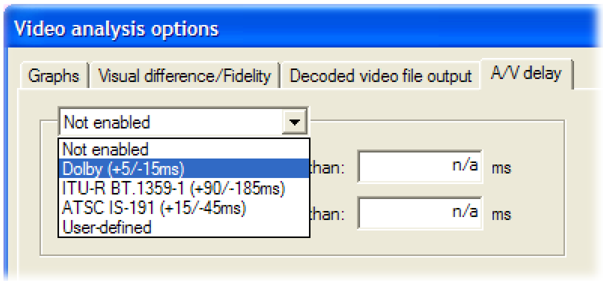


Not enabled

Audio must not lead video by more than: n/a ms

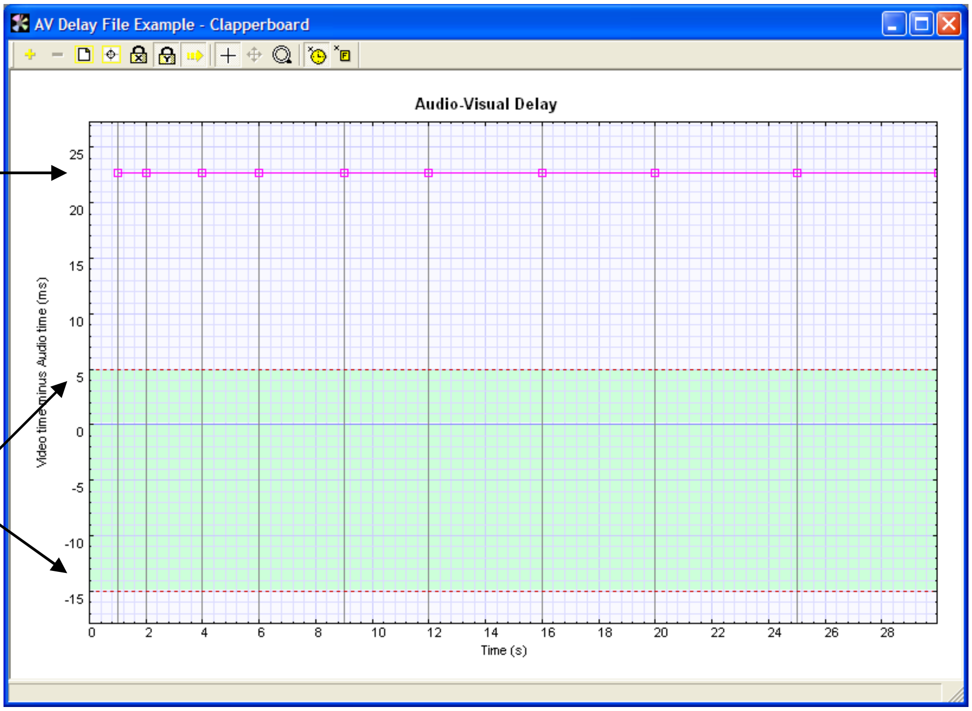
Audio must not lag video by more than: n/a ms

Select the required option from the drop-down list; the preset lead and lag values are shown in the list. If you want to set your own delay values, select User Defined and enter values in the lead and lag fields.



View A/V Delay Analysis…

Select Analysis > View A/V delay analysis… to display the A/V Delay graph. You will need to decode the stream to view the results.

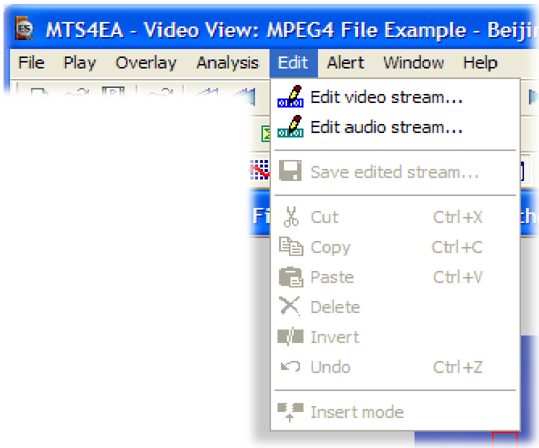


Audio/visual delay points (for example, 22.75 ms)

Limits for example, upper limit +5 ms lower limit -15 ms

The audio/visual limits you set (either preset or user-defined) are displayed in green. The actual delay is shown in red. In the preceding example, the limits are set at +5 ms and -15 ms. The example stream that has been decoded shows that the audio leads the video by 22.75 ms.

Edit Menu



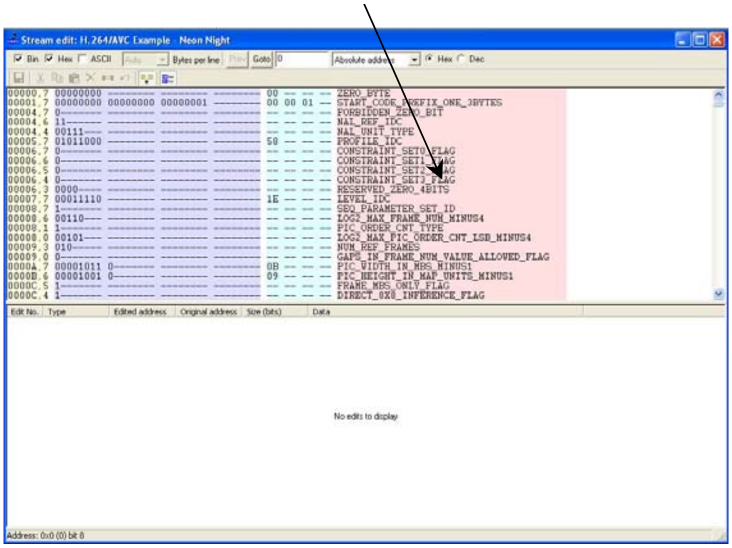
- The Bitstream editor facility allows you to:
  - Insert, delete and modify bits and bytes at any point in the bitstream
  - Copy sections of the bitstream
  - Rerun the analysis on the edited bitstream
  - See a change log of all edits that were made

Edit Audio/Video Stream...

When the Bitstream editor window is initially accessed, the following message will pop up advising you that the syntax will not be visible in this view until it is played (decoded):

“MTS4EA Compressed Video TS Analyzer: Warning
Syntax element labels will be available in the stream editor after the stream is decoded (video/audio is played).”

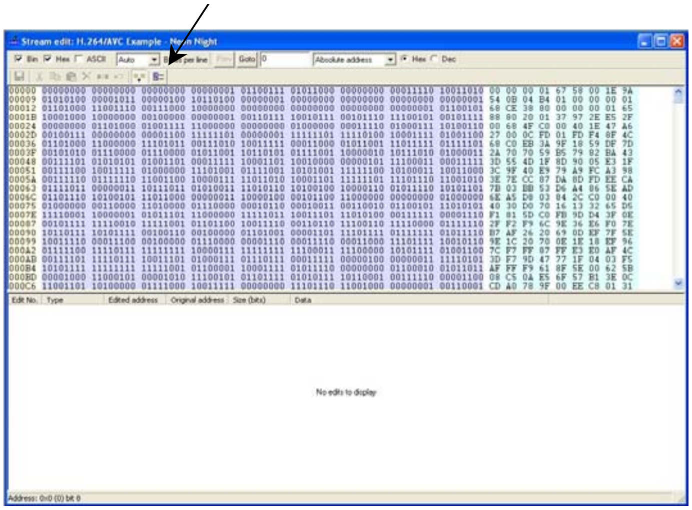
Once decoded, the syntax area will be populated.



```
00000.7 00000000 -------- -------- --------  00 -- -- --  ZERO_BYTE
00001.7 00000000 00000000 00000001 --------  00 00 01 --  START_CODE_PREFIX_ONE_3BYTES
00004.7 0------- -------- -------- --------  -- -- -- --  FORBIDDEN_ZERO_BIT
00004.6 11------ -------- -------- --------  -- -- -- --  NAL_REF_IDC
00004.4 00111--- -------- -------- --------  -- -- -- --  NAL_UNIT_TYPE
00005.7 01011000 -------- -------- --------  58 -- -- --  PROFILE_IDC
00006.7 0------- -------- -------- --------  -- -- -- --  CONSTRAINT_SET0_FLAG
00006.6 0------- -------- -------- --------  -- -- -- --  CONSTRAINT_SET1_FLAG
00006.5 0------- -------- -------- --------  -- -- -- --  CONSTRAINT_SET2_FLAG
00006.4 0------- -------- -------- --------  -- -- -- --  CONSTRAINT_SET3_FLAG
00006.3 0000---- -------- -------- --------  -- -- -- --  RESERVED_ZERO_4BITS
00007.7 00011110 -------- -------- --------  1E -- -- --  LEVEL_IDC
00008.7 1------- -------- -------- --------  -- -- -- --  SEQ_PARAMETER_SET_ID
00008.6 00110--- -------- -------- --------  -- -- -- --  LOG2_MAX_FRAME_NUM_MINUS4
00008.1 1------- -------- -------- --------  -- -- -- --  PIC_ORDER_CNT_TYPE
00008.0 00101--- -------- -------- --------  -- -- -- --  LOG2_MAX_PIC_ORDER_CNT_LSB_MINUS4
00009.3 010----- -------- -------- --------  -- -- -- --  NUM_REF_FRAMES
00009.0 0------- -------- -------- --------  -- -- -- --  GAPS_IN_FRAME_NUM_VALUE_ALLOWED_FLAG
0000A.7 00001011 0------- -------- --------  0B -- -- --  PIC_WIDTH_IN_MBS_MINUS1
0000B.6 00001001 0------- -------- --------  09 -- -- --  PIC_HEIGHT_IN_MAP_UNITS_MINUS1
0000C.5 1------- -------- -------- --------  -- -- -- --  FRAME_MBS_ONLY_FLAG
0000C.4 1------- -------- -------- --------  -- -- -- --  DIRECT_8X8_INFERENCE_FLAG
```

Address: 0x0 (0) bit 0

The pink area containing the syntax is displayed by default. If so desired, these labels can be removed from the display by clicking on the “Show labels” button:



```
00000 00000000 00000000 00000000 00000001 01100111 01011000 00000000 00011110 10011010 00 00 00 01 67 58 00 1E 9A
00009 01010100 00001011 00000100 10110100 00000001 00000000 00000000 00000000 00000001 54 0B 04 B4 01 00 00 00 01
00012 01101000 11001110 00111000 10000000 00000000 00000000 00000000 00000001 01100101 68 CE 38 80 00 00 00 01 65
0001B 10001000 10000000 00100000 00000001 00110111 10010111 00101110 11100101 00101111 88 80 20 01 37 97 2E E5 2F
00024 00000000 01101000 01001111 11000000 00000000 01000000 00011110 01000111 10100110 00 68 4F C0 00 40 1E 47 A6
0002D 00100111 00000000 00001100 11111101 00000001 11111101 11110100 10001111 01001100 27 00 0C FD 01 FD F4 8F 4C
00036 01101000 11000000 11101011 00111010 10011111 00011000 01011001 11011111 01111101 68 C0 EB 3A 9F 18 59 DF 7D
0003F 00101010 01110000 01110000 01011001 10110101 01111001 10000010 10111010 01000011 2A 70 70 59 B5 79 82 BA 43
00048 00111101 01010101 01001101 00011111 10001101 10010000 00000101 11100011 00011111 3D 55 4D 1F 8D 90 05 E3 1F
00051 00111100 10011111 01000000 11101001 01111001 10101001 11111100 10100011 10011000 3C 9F 40 E9 79 A9 FC A3 98
0005A 00111110 01111110 11001100 10000111 11011010 10001101 11111101 11101110 11001010 3E 7E CC 87 DA 8D FD EE CA
00063 01111011 00000011 10111011 01010011 11010110 10100100 10000110 01011110 10101101 7B 03 BB 53 D6 A4 86 5E AD
0006C 01101110 10100101 11011000 00000011 10000100 00101100 11000000 00000000 01000000 6E A5 D8 03 84 2C C0 00 40
00075 01000000 00110000 11010000 01110000 00010110 00010011 00110010 01100101 11010101 40 30 D0 70 16 13 32 65 D5
0007E 11110001 10000001 01011101 11000000 11111011 10011101 11010100 00111111 00001110 F1 81 5D C0 FB 9D D4 3F 0E
00087 00101111 11110010 11111001 01101100 10011110 00110110 11100110 11110000 01111110 2F F2 F9 6C 9E 36 E6 F0 7E
00090 10110111 10101111 00100110 00100000 01101001 00001101 11101111 01111111 01011110 B7 AF 26 20 69 0D EF 7F 5E
00099 10011110 00011100 00100000 01110000 00001110 00011110 00011000 11101111 10010110 9E 1C 20 70 0E 1E 18 EF 96
000A2 01111100 11110111 11111111 10000111 11111111 11100011 11100000 10101111 01001100 7C F7 FF 87 FF E3 E0 AF 4C
000AB 00111101 11110111 10011101 01000111 01110111 00011111 00000100 00000011 11110101 3D F7 9D 47 77 1F 04 03 F5
000B4 10101111 11111111 11111001 01100001 10001111 01011110 00000000 01100010 01011011 AF FF F9 61 8F 5E 00 62 5B
000BD 00001000 11000101 00001010 11100101 01101111 01010111 10110001 00111110 00001100 08 C5 0A E5 6F 57 B1 3E 0C
000C6 11001101 10100000 01111000 10011111 00000000 11101110 11001000 00000001 00110001 CD A0 78 9F 00 EE C8 01 31
```

Address: 0x0 (0) bit 0

The bottom half of the window - the change log - is empty until edits have been made to the stream.

Bitstream Editor Icons

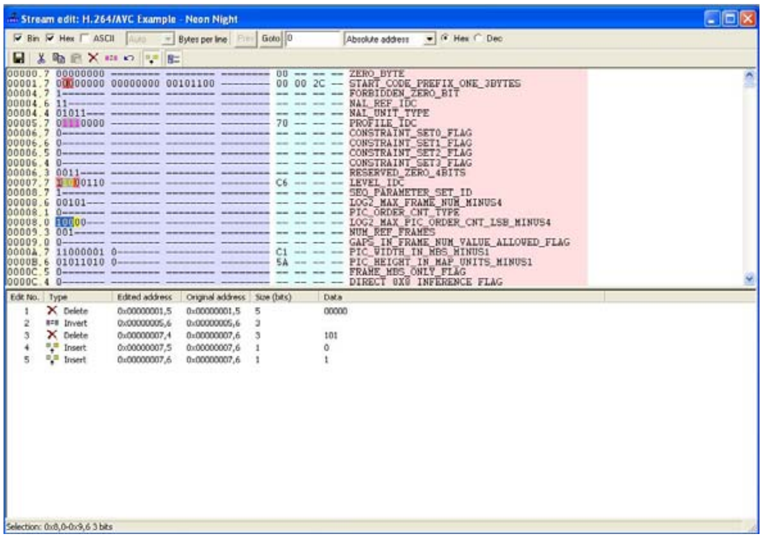
The functions of the remaining icons in the Bitstream editor view are outlined in the following table:

Table 6-6: Bitstream editor icons

Icon	Function
💾	The edited stream can be given a new name and saved
✂	Cut (Ctrl+X) becomes available only when part of the stream is highlighted
⎘	Copy (Ctrl+C) becomes available only when part of the stream is highlighted
📋	Paste (Ctrl+V) becomes available only when part of the stream is highlighted and copied to the clipboard
✕	Delete bits; becomes available only when part of the stream is highlighted
0⇄1	Inverts bits highlighted 0 becomes 1, 1 becomes 0; becomes available only when part of the stream is highlighted
↶	Undo last edit (Ctrl+Z)
0 01 ↑	Insert bits mode (this is enabled by default)
10 ⋯ 01 ⋯	Show labels (this is enabled by default

Change Log

As changes are made to the bitstream, the displayed text changes in color and the changes made appear as list items in the bottom half of the window. As in most of the views in MTS4EA, it is possible to highlight and double-click on any one of these items to be taken to the corresponding area in the relevant view. In this case, the bitstream view that occupies the top half of the window.



```
00000.7 00000000 -------- -------- -------- 00 -- -- -- ZERO_BYTE
00001.7 00000000 00000000 00101100 -------- 00 00 2C -- START_CODE_PREFIX_ONE_3BYTES
00004.7 1------- -------- -------- -------- -- -- -- -- FORBIDDEN_ZERO_BIT
00004.6 11------ -------- -------- -------- -- -- -- -- NAL_REF_IDC
00004.4 01011--- -------- -------- -------- -- -- -- -- NAL_UNIT_TYPE
00005.7 01110000 -------- -------- -------- 70 -- -- -- PROFILE_IDC
00006.7 0------- -------- -------- -------- -- -- -- -- CONSTRAINT_SET0_FLAG
00006.6 0------- -------- -------- -------- -- -- -- -- CONSTRAINT_SET1_FLAG
00006.5 0------- -------- -------- -------- -- -- -- -- CONSTRAINT_SET2_FLAG
00006.4 0------- -------- -------- -------- -- -- -- -- CONSTRAINT_SET3_FLAG
00006.3 0011---- -------- -------- -------- -- -- -- -- RESERVED_ZERO_4BITS
00007.7 11000110 -------- -------- -------- C6 -- -- -- LEVEL_IDC
00008.7 1------- -------- -------- -------- -- -- -- -- SEQ_PARAMETER_SET_ID
00008.6 00101--- -------- -------- -------- -- -- -- -- LOG2_MAX_FRAME_NUM_MINUS4
00008.1 0------- -------- -------- -------- -- -- -- -- PIC_ORDER_CNT_TYPE
00008.0 10000--- -------- -------- -------- -- -- -- -- LOG2_MAX_PIC_ORDER_CNT_LSB_MINUS4
00009.3 001----- -------- -------- -------- -- -- -- -- NUM_REF_FRAMES
00009.0 0------- -------- -------- -------- -- -- -- -- GAPS_IN_FRAME_NUM_VALUE_ALLOWED_FLAG
0000A.7 11000001 0------- -------- -------- C1 -- -- -- PIC_WIDTH_IN_MBS_MINUS1
0000B.6 01011010 0------- -------- -------- 5A -- -- -- PIC_HEIGHT_IN_MAP_UNITS_MINUS1
0000C.5 0------- -------- -------- -------- -- -- -- -- FRAME_MBS_ONLY_FLAG
0000C.4 0------- -------- -------- -------- -- -- -- -- DIRECT_8X8_INFERENCE_FLAG
```

Edit No.	Type	Edited address	Original address	Size (bits)	Data
1	Delete	0x00000001,5	0x00000001,5	5	00000
2	Invert	0x00000005,6	0x00000005,6	3	
3	Delete	0x00000007,4	0x00000007,6	3	101
4	Insert	0x00000007,5	0x00000007,6	1	0
5	Insert	0x00000007,6	0x00000007,6	1	1

Selection: 0x8,0-0x9,6 3 bits

As can be seen in this example, each numbered change lists both the Edited and Original addresses. This is because edits will have a cumulative effect on the addresses of bits later in the stream. So, if a bit or bits are deleted or inserted, the addresses of the following bits are changed. Therefore, to avoid confusion, both the original address and the address following earlier deletions or insertions are given.

Undo Last Edit ↶ Ctrl+Z

The last edit made to the bitstream can be undone by clicking on this icon or by using the shortcut Ctrl+Z.

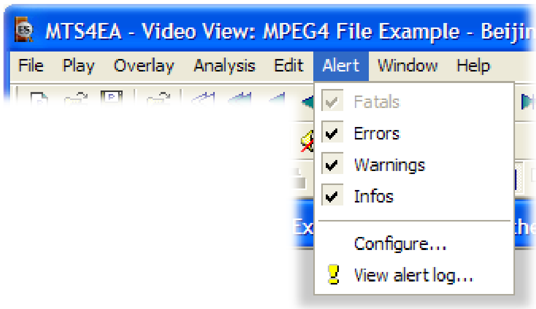
Saving the Edited Stream

Once all desired changes have been made, the stream can then be immediately played and reanalyzed. Alternatively, the stream can be saved under a new name and reopened later for analysis.

Edit Commands

The Edit menu options (Cut, Copy, Paste, Delete, Invert, Undo, Insert Mode) have the same functions as those described in Table 6-6, page 6–233.

Alert Menu



This menu controls the setting of the overall level of the popup alerts and gives access to configure individual alerts as enabled/disabled.

These popup alerts occur when the video is being decoded. Consequently:

- The popup alert appears when the frame concerned is decoded, but for bitstreams containing B-frames/B-VOPs the display order of frames/VOPs may be different

- When seeking backwards/forwards through a stream, the popup alerts will not appear if that part of the bitstream has already been decoded and is in the step-back buffer cache (see *Popup alerts in the step-back buffer* on page 6–242)

NOTE. *For some Alerts, extra information is provided about the section of the stream that has generated the error, and the standard if it is standard-specific. See* General codes used in Trace files and Alerts *in the* Bitstream Syntax Debugging *section for detailed information.*

Alert Levels

There are four different levels of alerts: Fatal, Error, Warning and Info (in order of decreasing severity). See *Description of Alert Levels* 6–246 for more information.

When a less severe alert is set, the more severe alerts are set automatically. (For example, setting Warnings on automatically sets Errors and Fatal on.)

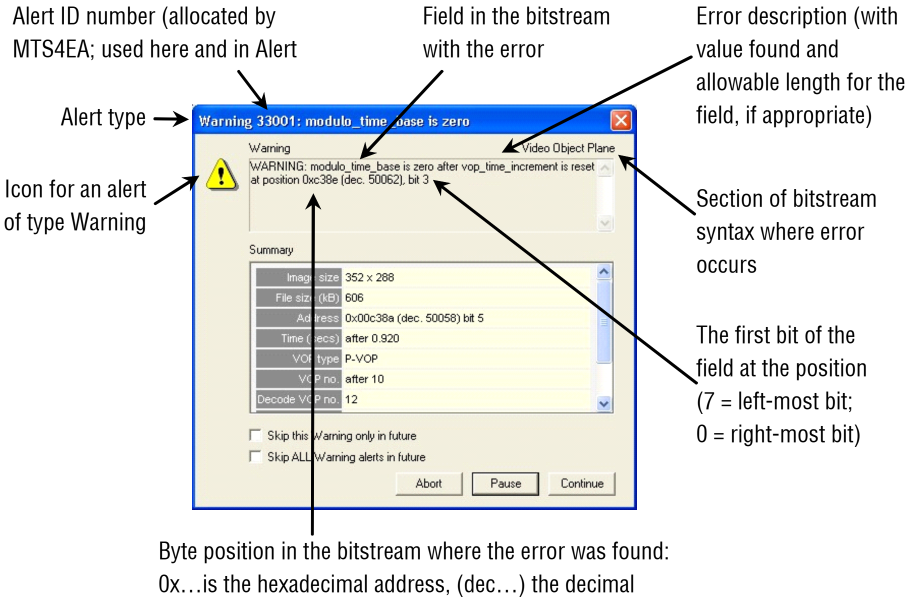
NOTE. *The Fatal alert is always set (and cannot be turned off).*
Eah time a video stream is opened in the MTS4EA, the alert level is set to Info (the strictest) and all Alerts are reenabled.

When an alert is triggered, a message box is displayed. At the same time, decoding is suspended.

The message box will provide you with the reason that decoding was interrupted - see the following section.

Explanation of Fatal/Error/Warning/Info Display

Information at Top of Alert Popup



Alert ID number (allocated by MTS4EA; used here and in Alert

Field in the bitstream with the error

Error description (with value found and allowable length for the field, if appropriate)

Alert type

Icon for an alert of type Warning

Section of bitstream syntax where error occurs

The first bit of the field at the position (7 = left-most bit; 0 = right-most bit)

Byte position in the bitstream where the error was found: 0x…is the hexadecimal address, (dec…) the decimal

Warning 33001: modulo_time_base is zero

Warning

Video Object Plane

WARNING: modulo_time_base is zero after vop_time_increment is reset at position 0xc38e (dec. 50062), bit 3

Summary

Image size	352 x 288
File size (kB)	606
Address	0x00c38a (dec. 50058) bit 5
Time (secs)	after 0.920
VOP type	P-VOP
VOP no.	after 10
Decode VOP no.	12

☐ Skip this Warning only in future

☐ Skip ALL Warning alerts in future

For bitstreams that contain B-VOPs/B-frames, it is vital to understand that the popup alert relates to the decode frame number, not the displayed frame number - see *Decode/display frame/VOP numbers in bitstreams with B-frames/B-VOPs* on page 6–240 for more information.

For all Fatal, Error, Warning and Info displays:

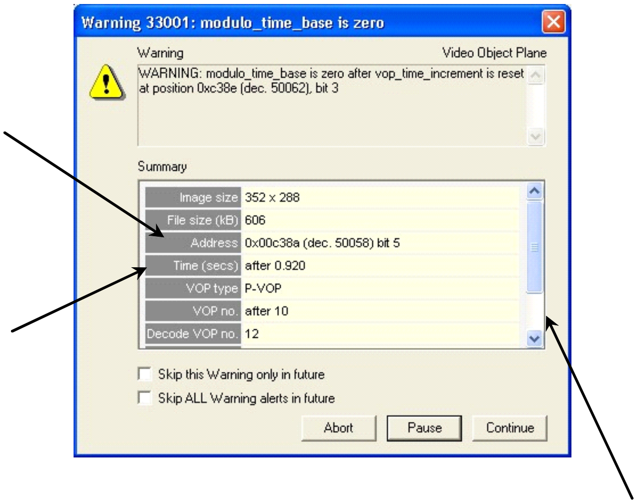
- The bitstream position of the error is the number of bytes from the beginning of the video bitstream, where the first byte is byte zero.
- The position is displayed in hexadecimal (0x...) and decimal (dec. ...).
- The bit start is the first bit of this bitstream field, where bit 7 is the most significant bit in a byte and bit 0 the least. This means that bit 7 is the first bit in each byte of the bitstream.

In each of the compression standards, there are many bitstream fields that give values that are least significant bit first (lsbf) and many others that are most significant bit first (msbf). There are also many others that are simply bit-patterns: they do not have a direct numeric value.

For msbf bitstream fields, numbering bit 7 as the first bit in each byte is correct. For lsbf bitstream fields, the bit order has to be reversed when calculating the given value . See *Explanation of bit/bitstart* in the *Bitstream Syntax Debugging* section for more information.

Information in Summary Box

The information provided in the Summary box is the same as that provided in the Summary tooltip; see *Video Summary Tooltip* Σ Ctrl+U on page 6–74 for a detailed explanation of the fields and provided data.



Warning 33001: modulo_time_base is zero

Warning — Video Object Plane

WARNING: modulo_time_base is zero after vop_time_increment is reset at position 0xc38e (dec. 50062), bit 3

Summary

Image size	352 x 288
File size (kB)	606
Address	0x00c38a (dec. 50058) bit 5
Time (secs)	after 0.920
VOP type	P-VOP
VOP no.	after 10
Decode VOP no.	12

☐ Skip this Warning only in future

☐ Skip ALL Warning alerts in future

Note: this is the address of the first bit in the frame/VOP, not the address of the error

In this example, the exact time of the error cannot be determined as the error occurs during the decoding of a frame/VOP header

Usually, there will be too many fields to all be displayed within the available space. In which case, a vertical scroll bar appears. Scroll down to see the remaining fields

Decode/display frame/VOP numbers in bitstreams with B-frames/B-VOPs. The popup alert relates to the Warnings and Errors that the MTS4EA finds during decoding.

This means that for bitstreams that contain B-frames/B-VOPs (where the decode order is different from the display order) it can appear that the popup alerts are out of sequence. In this case, it is very important to look carefully at the data in the Summary box, which shows the Decode and Display frame numbers.

As an example:

- Order of data in the bitstream (the decode order):

Decode frame number	1	2	3	4	5
Frame type	I	P	B	B	P
Error occurs in frame	a	b	c	d	e

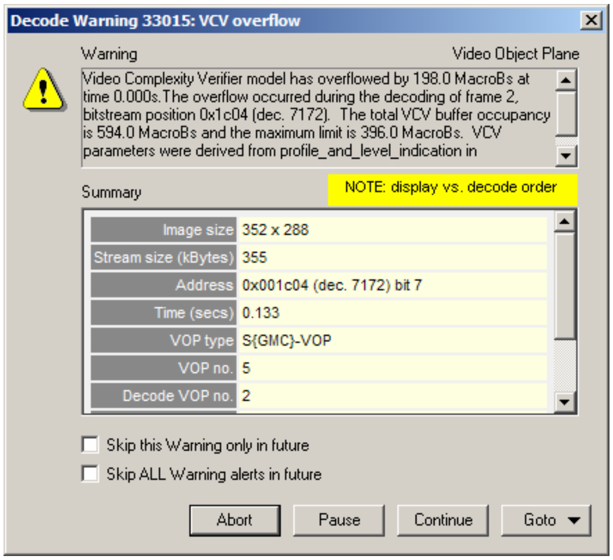
- Order of data in displayed:

Display frame number	-	1	2	3	4	5
Frame type	-	I	B	B	P	P
Error displayed in frame	-	a, b	c	d	-	e

NOTE: *Frame display delayed by one frame*

In these diagrams, it can be seen that for the first P-frame, the error associated with decoding it is displayed before it appears on-screen - at the same time as the first frame is displayed.

If this has happened, text in a yellow warning rectangle will appear in the popup alert:



Decode Warning 33015: VCV overflow

Warning

Video Object Plane

Video Complexity Verifier model has overflowed by 198.0 MacroBs at time 0.000s.The overflow occurred during the decoding of frame 2, bitstream position 0x1c04 (dec. 7172). The total VCV buffer occupancy is 594.0 MacroBs and the maximum limit is 396.0 MacroBs. VCV parameters were derived from profile_and_level_indication in

Summary

NOTE: display vs. decode order

Image size	352 x 288
Stream size (kBytes)	355
Address	0x001c04 (dec. 7172) bit 7
Time (secs)	0.133
VOP type	S{GMC}-VOP
VOP no.	5
Decode VOP no.	2

□ Skip this Warning only in future

□ Skip ALL Warning alerts in future

The difference in decode versus display order:

- Can occur in any bitstream which has B-frames/B-VOPs
- And can happen in any frame/VOP in the bitstream concerned, where the frame/VOP is a B-frame/B-VOP

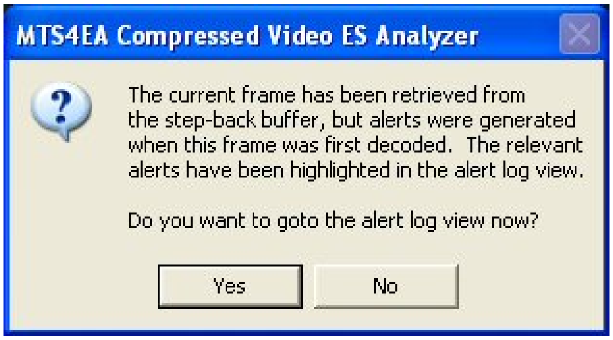
NOTE. *Note that the yellow warning can also appear in other circumstances, where the alert has occurred some time after a displayed frame.*

Popup alerts in the step-back buffer. When a stream is decoded, popup alerts may occur. The results of the decoding are stored in a cache, the Step-back buffer - see *General* on page 6–60 for more information.

It is possible to step forward/backward over the frames where these alerts occur. To avoid a continual repetition of the popups (and potentially many such alerts appearing), the popup alerts are not displayed when seeking forward/backward in these situations.

If one or more popup alerts would have occurred, then this is indicated in one of two ways:

- There may be a popup message, at most once per seek forward or backward:

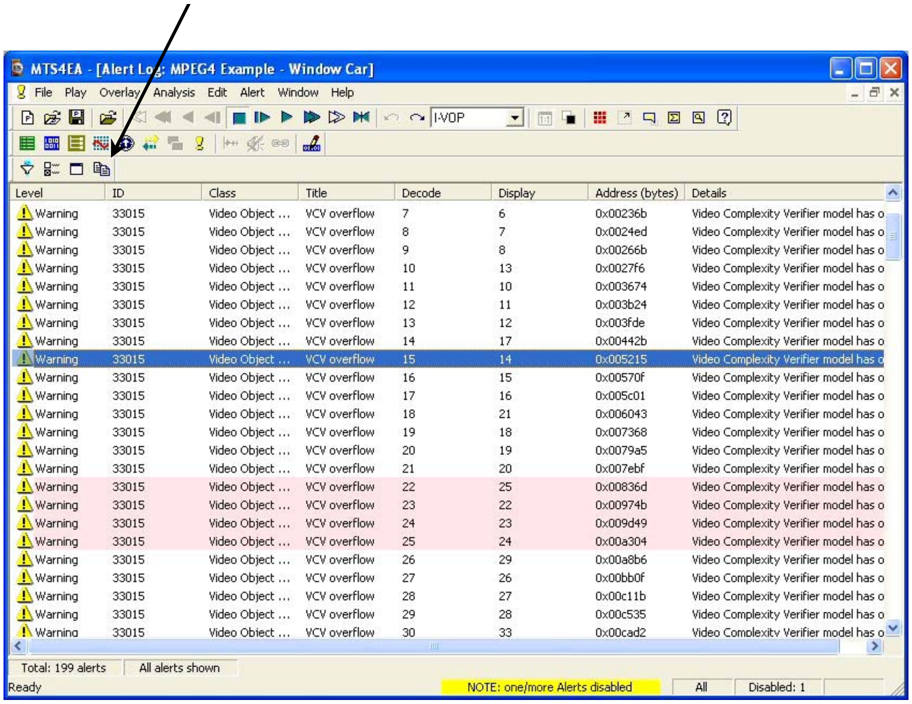


The current frame has been retrieved from the step-back buffer, but alerts were generated when this frame was first decoded. The relevant alerts have been highlighted in the alert log view.

Do you want to goto the alert log view now?

- If the Alert log window is open, the alerts that would have popped-up are highlighted in the Alert log window (see *View Alert Log* on page 6–251 for more information)

Copy alert details to the Windows clipboard. The details of an alert can be copied using `Ctrl+C` or the copy icon.



MTS4EA - [Alert Log: MPEG4 Example - Window Car]
File Play Overlay Analysis Edit Alert Window Help
I-VOP

Level	ID	Class	Title	Decode	Display	Address (bytes)	Details
Warning	33015	Video Object ...	VCV overflow	7	6	0x00236b	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	8	7	0x0024ed	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	9	8	0x00266b	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	10	13	0x0027f6	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	11	10	0x003674	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	12	11	0x003b24	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	13	12	0x003fde	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	14	17	0x00442b	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	15	14	0x005215	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	16	15	0x00570f	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	17	16	0x005c01	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	18	21	0x006043	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	19	18	0x007368	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	20	19	0x0079a5	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	21	20	0x007ebf	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	22	25	0x00836d	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	23	22	0x00974b	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	24	23	0x009d49	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	25	24	0x00a304	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	26	29	0x00a8b6	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	27	26	0x00bb0f	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	28	27	0x00c11b	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	29	28	0x00c535	Video Complexity Verifier model has o
Warning	33015	Video Object ...	VCV overflow	30	33	0x00cad2	Video Complexity Verifier model has o

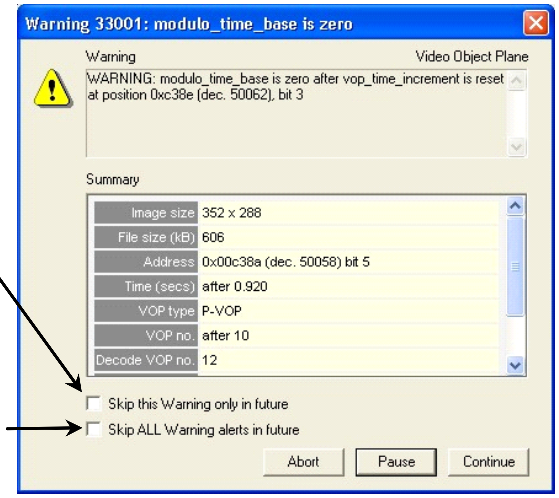
Total: 199 alerts | All alerts shown

Ready | NOTE: one/more Alerts disabled | All | Disabled: 1

Check box options.

Check this box to prevent seeing further alert(s) for this specific warning. Other Warnings, Errors and Fatals will still trigger an alert

Check this box to prevent seeing further alert(s) for any alert level



Warning 33001: modulo_time_base is zero

Warning | Video Object Plane

WARNING: modulo_time_base is zero after vop_time_increment is reset at position 0xc38e (dec. 50062), bit 3

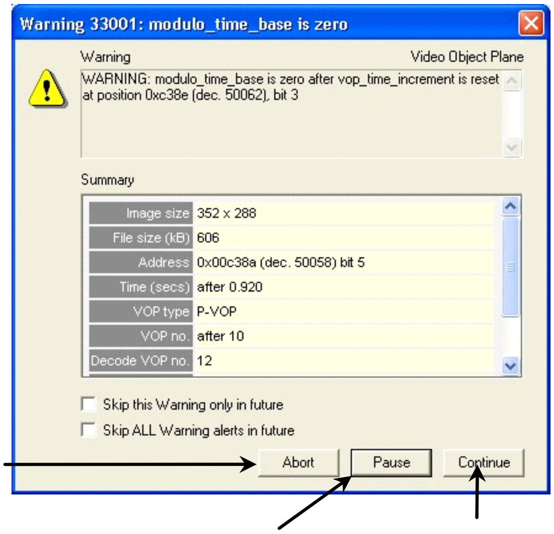
Summary

Image size	352 x 288
File size (kB)	606
Address	0x00c38a (dec. 50058) bit 5
Time (secs)	after 0.920
VOP type	P-VOP
VOP no.	after 10
Decode VOP no.	12

☐ Skip this Warning only in future

☐ Skip ALL Warning alerts in future

Abort, Pause, Continue.



Warning 33001: modulo_time_base is zero

Warning — Video Object Plane

WARNING: modulo_time_base is zero after vop_time_increment is reset at position 0xc38e (dec. 50062), bit 3

Summary

Image size	352 x 288
File size (kB)	606
Address	0x00c38a (dec. 50058) bit 5
Time (secs)	after 0.920
VOP type	P-VOP
VOP no.	after 10
Decode VOP no.	12

☐ Skip this Warning only in future

☐ Skip ALL Warning alerts in future

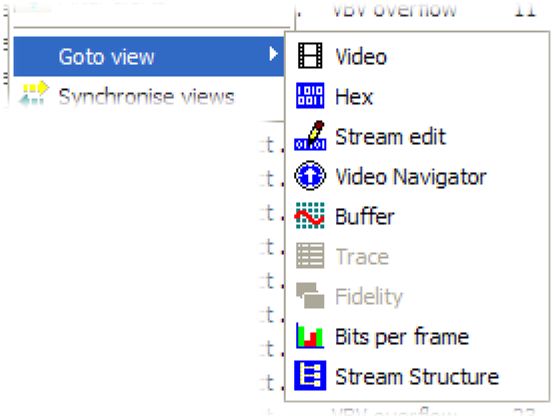
Abort stops the decoding immediately; at the end of the current alert, stream in Stop mode

Pause keeps decoding to the end of the current frame/VOP; thereafter stream enters Stop mode

Continue decoding (in normal/fast/step modes)

NOTE. *Abort can also be requested by pressing the Esc key.*

Goto button. This takes the focus to the position in the selected view that corresponds most closely to the location of the error. See *Synchronized Views/Navigating the Views* on page 6–7 for more information.



In these views, selecting Goto takes the focus to the following locations:

View	Location taken to	
Video	If the error can be located to within a particular MacroBlock…	…to the MacroBlock containing the error (highlighted with a yellow/black dotted square)
	If not (the error is in a frame header or file header)…	…to the top of the frame that immediately follows the error, indicated by a yellow/black dotted rectangle around the first row of MacroBlocks
Hex	The byte where the error is	
Stream edit	The byte where the error is	
Alert log	The log of alert popups that have occurred (not those filtered/not shown)	
Navigator	The frame containing the error or the frame that immediately follows the error (if the error is in a header)	
Buffer	The frame containing the error or the frame that immediately follows the error (if the error is in a header)	
Trace	The syntax element in the Interpret or Parse bitstream file corresponding to the location where the error has been found	
Fidelity	Graph of fidelity analysis (such as PSNR)	
Audio waveform	Graph of audio stream: Y-axis is normalized amplitude; X-axis is time (seconds)	
Bits per frame	Bits per frame graph – the frame containing the error	
Stream Structure	Stream structure – highlight node containing error	

Description of Alert Levels

Fatal

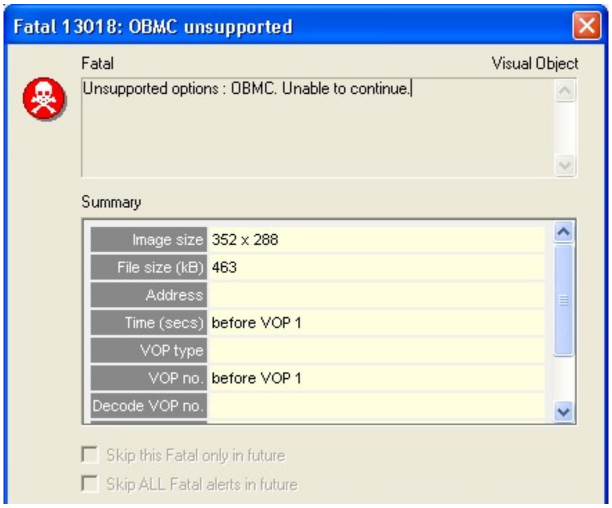
A Fatal error occurs when there is something encountered in the bitstream that cannot be decoded by the MTS4EA.

This selection cannot be turned off, and the decoder cannot continue after a Fatal error.

Fatal errors in the MTS4EA are commonly caused by:

- Severe errors in the syntax of the bitstream, which make it completely unintelligible to MTS4EA
- Options being selected in the video bitstream that are not yet supported by MTS4EA

For example, the Fatal error below was caused by the OBMC flag in an MPEG-4 bitstream being set (on), that the video bitstream in question is using overlapped block motion compensation. (OBMC = Overlapped Block Motion Compensation: although it is in the current MPEG-4 standard, none of the MPEG-4 profiles or levels defined currently support OBMC.)



Fatal 13018: OBMC unsupported

Fatal

Visual Object

Unsupported options : OBMC. Unable to continue.

Summary

Image size	352 x 288
File size (kB)	463
Address	
Time (secs)	before VOP 1
VOP type	
VOP no.	before VOP 1
Decode VOP no.	

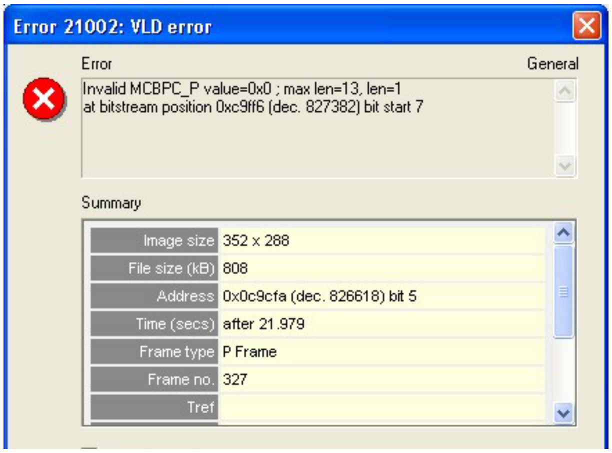
□ Skip this Fatal only in future

□ Skip ALL Fatal alerts in future

Error

When this item is selected, the decoder will stop at any point where there is an Error in the bitstream.

These occur commonly where out-of-range values have been used in the bitstream, or invalid codes (the example below in an H.263 stream):



Error 21002: VLD error

Error

General

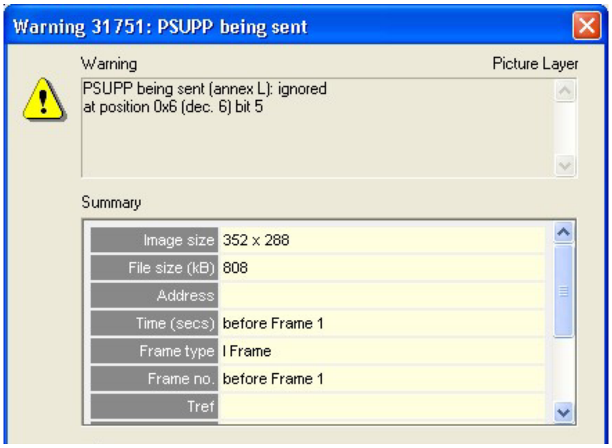
Invalid MCBPC_P value=0x0 ; max len=13, len=1
at bitstream position 0xc9ff6 (dec. 827382) bit start 7

Summary

Image size	352 x 288
File size (kB)	808
Address	0x0c9cfa (dec. 826618) bit 5
Time (secs)	after 21.979
Frame type	P Frame
Frame no.	327
Tref	

Warning

When selected, this item will provide a number of warnings that indicate behavior that is not correct, but does not necessarily damage the decoding of the bitstream. Items such as too many leading zeros before picture start codes, temporal references not set, and MacroBlocks not updated in Intra mode will cause such warnings (the example below in an H.263 stream):



Warning 31751: PSUPP being sent

Warning

Picture Layer

PSUPP being sent (annex L): ignored
at position 0x6 (dec. 6) bit 5

Summary

Image size	352 x 288
File size (kB)	808
Address	
Time (secs)	before Frame 1
Frame type	I Frame
Frame no.	before Frame 1
Tref	

Info

When checked, this item will provide information popups if relevant:

- It may be that the behavior is within the acceptable bounds of the standard concerned (and the relevant Profile/Level), but perhaps is used in a non-recommended manner or at the limits

- It is felt that additional information would be useful

Configure Alerts

This determines which specific alerts are enabled/disabled.

NOTE. *The overall alert level is set from the Alerts menu options Errors, Warnings and Infos, and this the enabling of specific alerts but not the disabling of specific alerts.*

This means, for example:

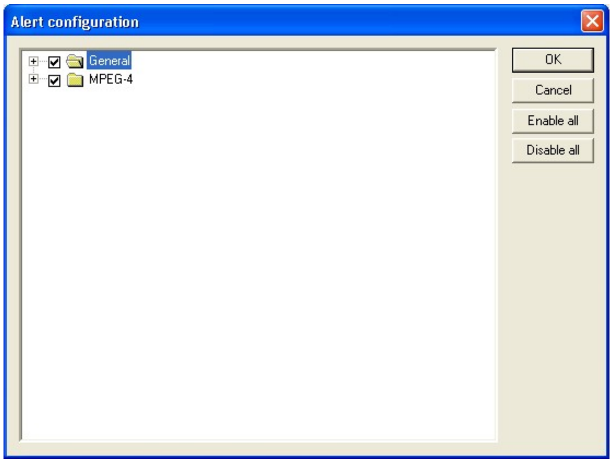
- That a specific Warning alert can be enabled/disabled, but trigger a Warning alert if the overall alert level is set to Errors or Fatal only. This is indicated in the status bar: see *Status Bar Indication of Alert Status* on page 6–258

- But if the overall alert level is Info and an individual Error alert is disabled, then an alert will not be triggered for this individual error

- Examples of different combinations of overall and individual alerts and the consequential status bar displays are given under *Status Bar Indication of Alert Status* on page 6–258

NOTE. *Depending upon the standard, all the alerts that MTS4EA tests for are not necessarily currently available on-screen to be individually enabled/ disabled. Access to more of these will be provided in future versions of MTS4EA.*

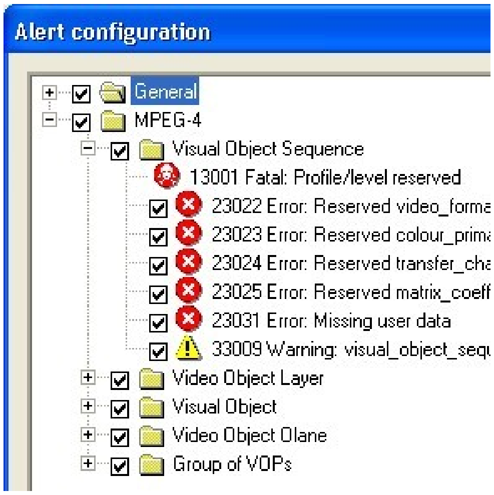
Enable/disable Specific Alerts



This displays two folders at the top level:

- General alerts, which can apply regardless of the standard concerned
- A standard folder (MPEG-4, H.263) that contains alerts specific to the standard concerned

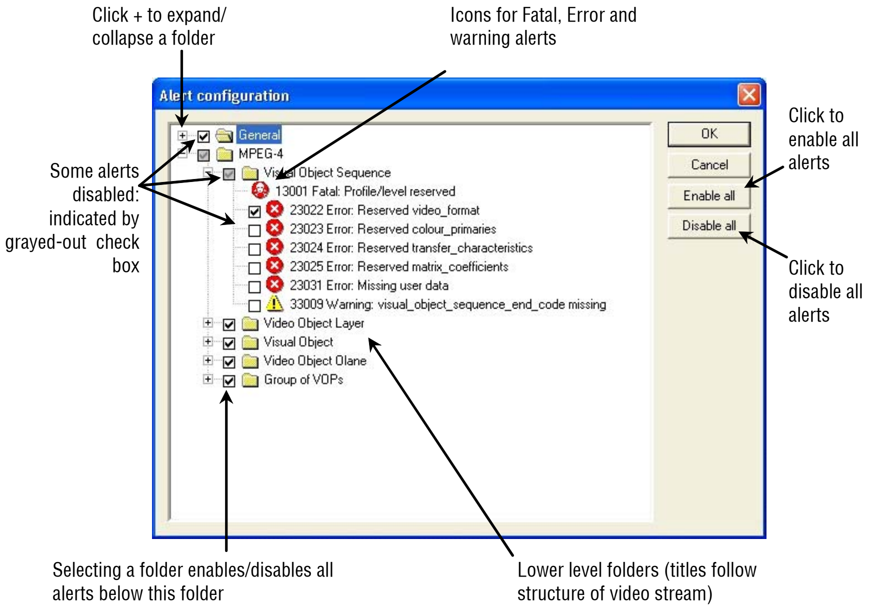
The + next to each folder can be clicked to reveal lower levels:



☑ General
☑ MPEG-4
☑ Visual Object Sequence
13001 Fatal: Profile/level reserved
☑ 23022 Error: Reserved video_forma
☑ 23023 Error: Reserved colour_prima
☑ 23024 Error: Reserved transfer_cha
☑ 23025 Error: Reserved matrix_coeff
☑ 23031 Error: Missing user data
☑ 33009 Warning: visual_object_sequ
☑ Video Object Layer
☑ Visual Object
☑ Video Object Olane
☑ Group of VOPs

Elements of the Alert Configuration Window

NOTE. *This is used to configure the individual alerts. Even if an alert is enabled here, if the overall alert level is set higher, the alert will not trigger a popup.*



⊞ ☑ General
⊟ ▣ MPEG-4
⊟ ▣ Visual Object Sequence
13001 Fatal: Profile/level reserved
☑ 23022 Error: Reserved video_format
□ 23023 Error: Reserved colour_primaries
□ 23024 Error: Reserved transfer_characteristics
□ 23025 Error: Reserved matrix_coefficients
□ 23031 Error: Missing user data
□ 33009 Warning: visual_object_sequence_end_code missing
⊞ ☑ Video Object Layer
⊞ ☑ Visual Object
⊞ ☑ Video Object Olane
⊞ ☑ Group of VOPs

Click + to expand/ collapse a folder

Icons for Fatal, Error and warning alerts

Some alerts disabled: indicated by grayed-out check box

Click to enable all alerts

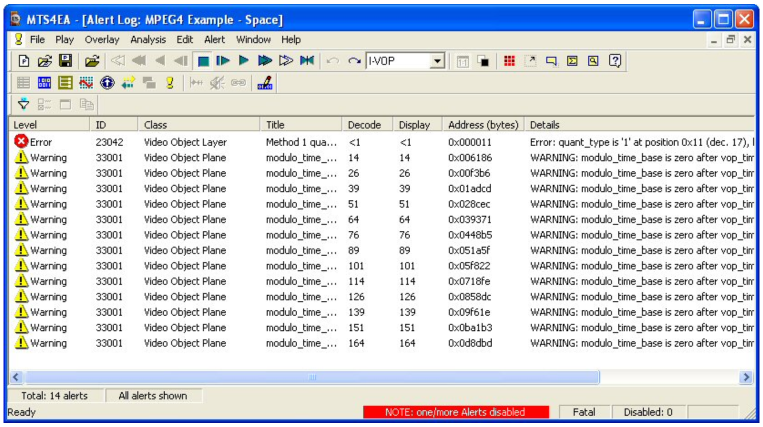
Click to disable all alerts

Selecting a folder enables/disables all alerts below this folder

Lower level folders (titles follow structure of video stream)

View Alert Log

The Alert log keeps a list of the popup alerts that have occurred during the decoding of the bitstream:



Level	ID	Class	Title	Decode	Display	Address (bytes)	Details
Error	23042	Video Object Layer	Method 1 qua…	<1	<1	0x000011	Error: quant_type is '1' at position 0x11 (dec. 17), I
Warning	33001	Video Object Plane	modulo_time_…	14	14	0x006186	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	26	26	0x00f3b6	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	39	39	0x01adcd	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	51	51	0x028cec	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	64	64	0x039371	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	76	76	0x0448b5	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	89	89	0x051a5f	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	101	101	0x05f822	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	114	114	0x0718fe	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	126	126	0x0858dc	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	139	139	0x09f61e	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	151	151	0x0ba1b3	WARNING: modulo_time_base is zero after vop_tim
Warning	33001	Video Object Plane	modulo_time_…	164	164	0x0d8dbd	WARNING: modulo_time_base is zero after vop_tim

Total: 14 alerts | All alerts shown

Ready | NOTE: one/more Alerts disabled | Fatal | Disabled: 0

The Alert log window icons and column titles are as follows:

Alert filter icon . See *Showing/Hiding Alerts (Alert Filter)* on page 6–253

Show/hide video alerts .

Show/hide audio alerts .

Configure alerts icon . Clicking on this icon opens the Alert configuration dialog box - see *Configure Alerts* on page 6–248. If this icon is clicked, then the Alert configuration dialog box is opened with the error selected in the Alert log window highlighted in the Alert configuration dialog box. If this icon is grayed out, it can be enabled by selecting any alert that is shown.

View alert details icon . See *View details of the alerts* on page 6–254. If this icon is grayed out, it can be enabled by selecting any alert that is shown.

Copy . This icon becomes available only once an alert is highlighted. This function allows the selected alert to be copied to the Windows clipboard for pasting into any Windows application. Copying can also be done by pressing `Ctrl+C`.

Level.. The error level: Fatal or Error or Warning or Info.

ID. The unique ID number assigned to this error type within MTS4EA.

Class. The area within the hierarchy of the compressed standard; see the screenshot above which shows an error in the Video Object Layer and seven warnings in the Video Object Plane within the MPEG-4 bitstream. The classes vary with the video standards.

Title. The title assigned to this error type.

Decode. The number of the decoded frame in which this error was found.

See *Decode/display frame/VOP numbers in bitstreams with B-frames/B-VOPs* on page 6–240 for more information on decoded versus displayed frame numbers, in bitstreams that include B-frames/B-VOPs.

NOTE. *If <1 is displayed, the error occurred before the decode of the first frame - the error occurred in the header.*

Display. The number of the displayed frame in which this error is shown.

See *Decode/display frame/VOP numbers in bitstreams with B-frames/B-VOPs* on page 6–240 for more information on decoded versus displayed frame numbers, in bitstreams that include B-frames/B-VOPs.

NOTE. *If <1 is displayed, this means that the error has occurred before the decode of the first frame, the error has occurred in the header.*

Address (bytes). The byte address in the bitstream where this error was found.

NOTE. *For video bitstreams that have been extracted from container files (for example, MP4 files, 3GPP files, MPEG-2 Packet Streams) this address is the byte address in the extracted video stream, not in the overall container file.*

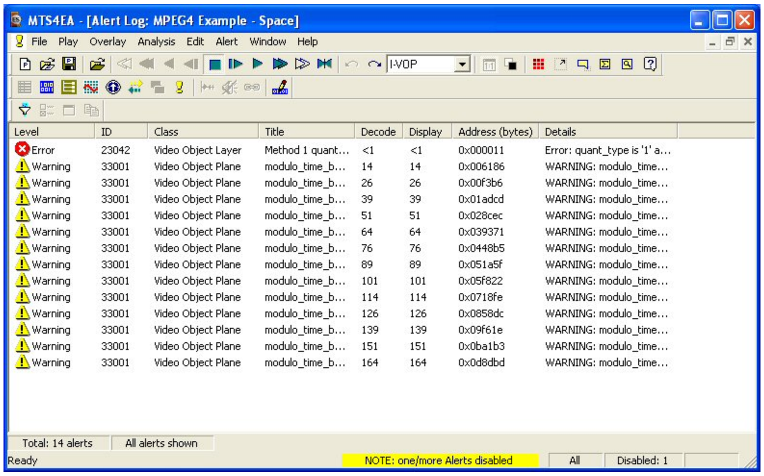
Details. This is the description text that is shown when the popup alert is displayed.

Showing/Hiding Alerts (Alert Filter)

In some cases there can be hundreds or thousands of alerts. These might be known issues and you do not want to list them in the Alert log window (because they would clutter the window unnecessarily).

Therefore, the alerts that are shown in the Alert log window can be filtered. The filtering is based on the setting of the alert popup enable in the Alert configuration (see *Configure Alerts* on page 6–248 regarding the Alert configuration dialog box):

- If the filter icon is out, all alerts are shown, regardless of whether they are enabled for a popup alert:



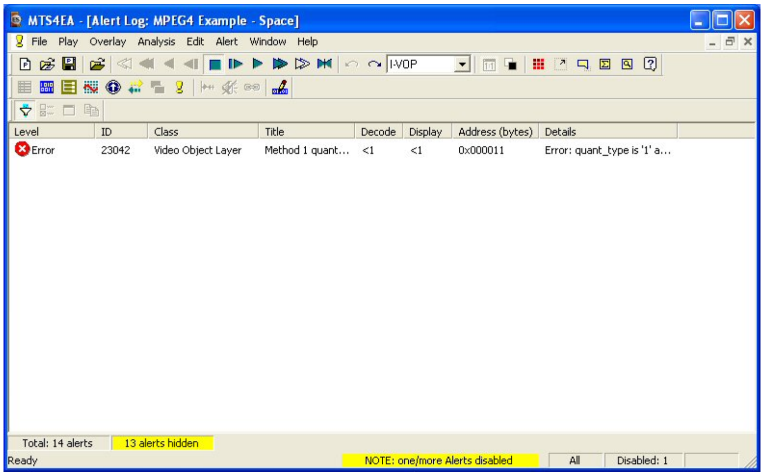
MTS4EA - [Alert Log: MPEG4 Example - Space]
File Play Overlay Analysis Edit Alert Window Help
I-VOP

Level	ID	Class	Title	Decode	Display	Address (bytes)	Details
Error	23042	Video Object Layer	Method 1 quant…	<1	<1	0x000011	Error: quant_type is '1' a…
Warning	33001	Video Object Plane	modulo_time_b…	14	14	0x006186	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	26	26	0x00f3b6	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	39	39	0x01adcd	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	51	51	0x028cec	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	64	64	0x039371	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	76	76	0x0448b5	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	89	89	0x051a5f	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	101	101	0x05f822	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	114	114	0x0718fe	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	126	126	0x0858dc	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	139	139	0x09f61e	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	151	151	0x0ba1b3	WARNING: modulo_time…
Warning	33001	Video Object Plane	modulo_time_b…	164	164	0x0d8dbd	WARNING: modulo_time…

Total: 14 alerts | All alerts shown

Ready | NOTE: one/more Alerts disabled | All | Disabled: 1

- If the filter icon is pushed in , only those alerts that are enabled are shown:



MTS4EA - [Alert Log: MPEG4 Example - Space]

Level	ID	Class	Title	Decode	Display	Address (bytes)	Details
Error	23042	Video Object Layer	Method 1 quant...	<1	<1	0x000011	Error: quant_type is '1' a...

Total: 14 alerts | 13 alerts hidden

Ready | NOTE: one/more Alerts disabled | All | Disabled: 1

(In this example, 13 alerts are hidden out of the total of 14 alerts)

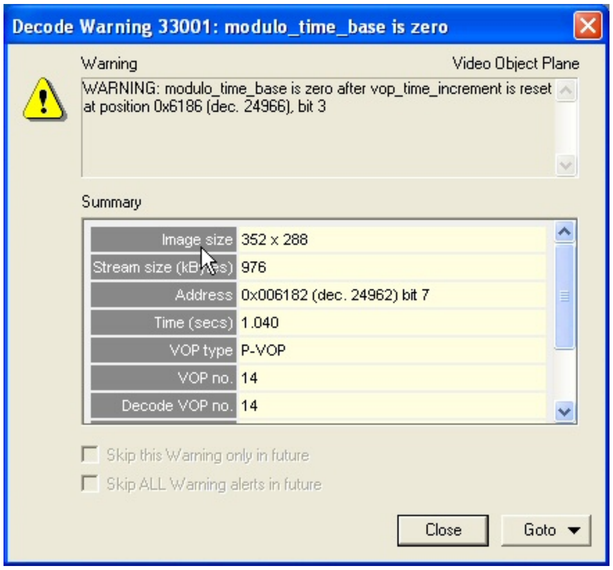
The Alert log filter can also be turned on/off by right-clicking and selecting the option from the popup menu.

NOTE. *When the filter icon is grayed out, all the alerts are enabled for display, so that there are no alerts to filter.*

View details of the alerts. This can be done in a number of ways:

- Double-click an alert in the Alert log window
- Single-click an alert, right-clicking to get the popup menu, and then click View alert details
- Select an alert by single-clicking an alert, and then clicking the icon

When this is done, a display very similar to the original popup alert display is shown:



Decode Warning 33001: modulo_time_base is zero

Warning — Video Object Plane

WARNING: modulo_time_base is zero after vop_time_increment is reset at position 0x6186 (dec. 24966), bit 3

Summary

Image size	352 x 288
Stream size (kBytes)	976
Address	0x006182 (dec. 24962) bit 7
Time (secs)	1.040
VOP type	P-VOP
VOP no.	14
Decode VOP no.	14

□ Skip this Warning only in future

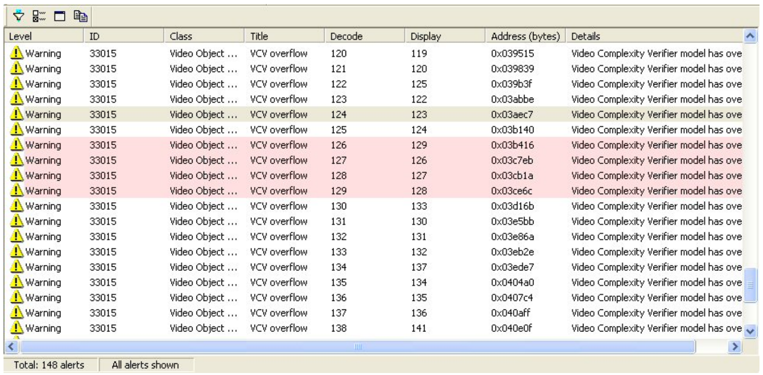
□ Skip ALL Warning alerts in future

Alert highlighting when seeking forward/backward through video in the step-back buffer. The MTS4EA buffers video (and the data associated with each video frame) in a step-back buffer or cache, see *General* on page 6–60 more information.

When stepping backward/forward through video that is held in the step-back buffer, to avoid continually having known alerts appear. The MTS4EA only highlights the alerts relevant to the area being skipped in the Alert log.

For example, if you skip forward 100 frames, and there would normally be four popup alerts during the decoding of these frames, the MTS4EA will highlight the four alerts in the Alert log. (If the alerts have been skipped and are hidden in the Alert log, as indicated by the status bar of the Alert log, the MTS4EA cannot highlight the alerts.)

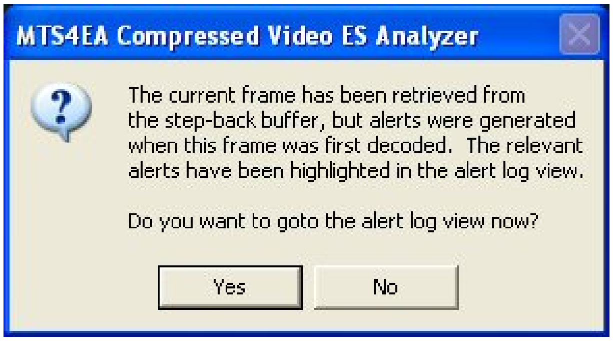
An example of the highlighting of alerts is shown in the following screenshot:



Level	ID	Class	Title	Decode	Display	Address (bytes)	Details
⚠ Warning	33015	Video Object ...	VCV overflow	120	119	0x039515	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	121	120	0x039839	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	122	125	0x039b3f	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	123	122	0x03abbe	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	124	123	0x03aec7	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	125	124	0x03b140	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	126	129	0x03b416	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	127	126	0x03c7eb	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	128	127	0x03cb1a	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	129	128	0x03ce6c	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	130	133	0x03d16b	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	131	130	0x03e5bb	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	132	131	0x03e86a	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	133	132	0x03eb2e	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	134	137	0x03ede7	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	135	134	0x0404a0	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	136	135	0x0407c4	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	137	136	0x040aff	Video Complexity Verifier model has ove
⚠ Warning	33015	Video Object ...	VCV overflow	138	141	0x040e0f	Video Complexity Verifier model has ove

Total: 148 alerts | All alerts shown

If the Alert log is closed, then the MTS4EA displays a warning, as shown in the following figure:



MTS4EA Compressed Video ES Analyzer

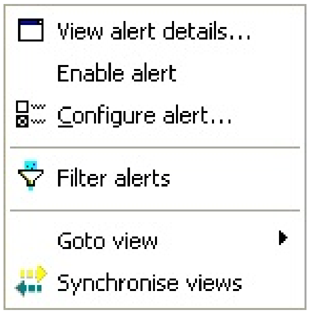
The current frame has been retrieved from the step-back buffer, but alerts were generated when this frame was first decoded. The relevant alerts have been highlighted in the alert log view.

Do you want to goto the alert log view now?

This is displayed only once for all the Alerts between the relevant frames.

Right-click popup menu

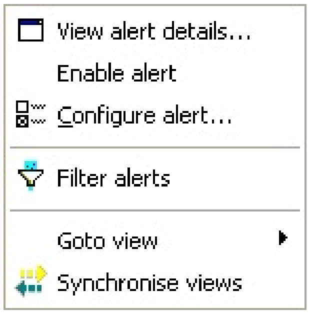
The available functions in the Alert log can be selected by right-clicking and selecting the function from the popup menu:



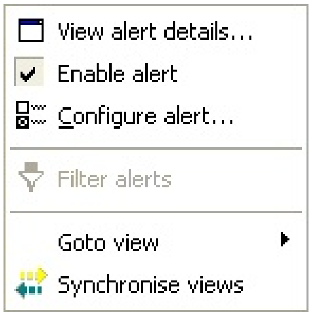
The functions of this menu are:

View alert details.... See *View details of the alerts* on page 6–254.

Enable alert. An individual alert can be enabled or disabled by clicking on this menu selection:



Alert disabled



Alert enabled

NOTE. *This allows an individual alert to be enabled/disabled from this menu, without having to open the Alert configuration dialog box.*

Configure alert... See *Configure Alerts* on page 6–248.

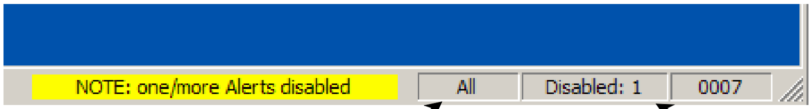
Filter alerts. See *Showing/Hiding Alerts (Alert Filter)* on page 6–253.

Go to view. See *Synchronized Views/Navigating the Views* on page 6–7.

Synchronize views. See *Synchronized Views/Navigating the Views* on page 6–7.

Status Bar Indication of Alert Status

The status bar displays the status of the alerts.



NOTE: one/more Alerts disabled | All | Disabled: 1 | 0007

Overall alert level enabled,
- All = all levels, including Infos
- Warning = Warnings, Errors, Fatals
- Error = Errors and Fatals
- Fatal = Fatals only

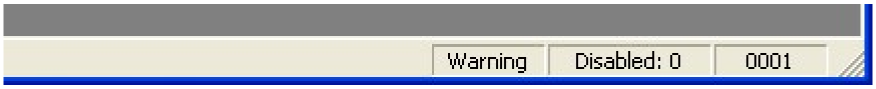
Number of specific alerts disabled

One or more alerts disabled

When one or more alerts are disabled, either because:

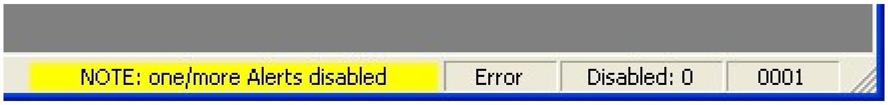
- The overall alert level does not include Warnings, Errors and Fatals
- One or more of the specific alerts have been disabled

The status bar changes as shown:



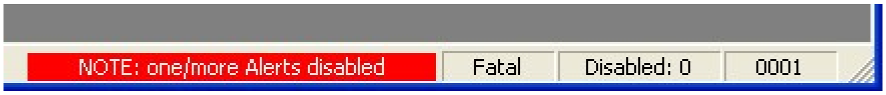
Warning | Disabled: 0 | 0001

Overall alert level: Infos disabled (all other levels enabled)



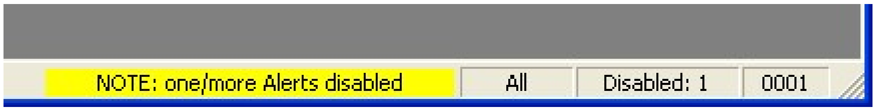
NOTE: one/more Alerts disabled | Error | Disabled: 0 | 0001

Overall alert level: Warnings disabled (Errors and Fatals enabled)



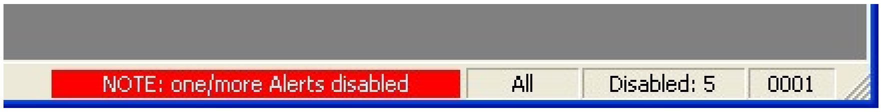
NOTE: one/more Alerts disabled | Fatal | Disabled: 0 | 0001

Overall alert level: Errors disabled



NOTE: one/more Alerts disabled | All | Disabled: 1 | 0001

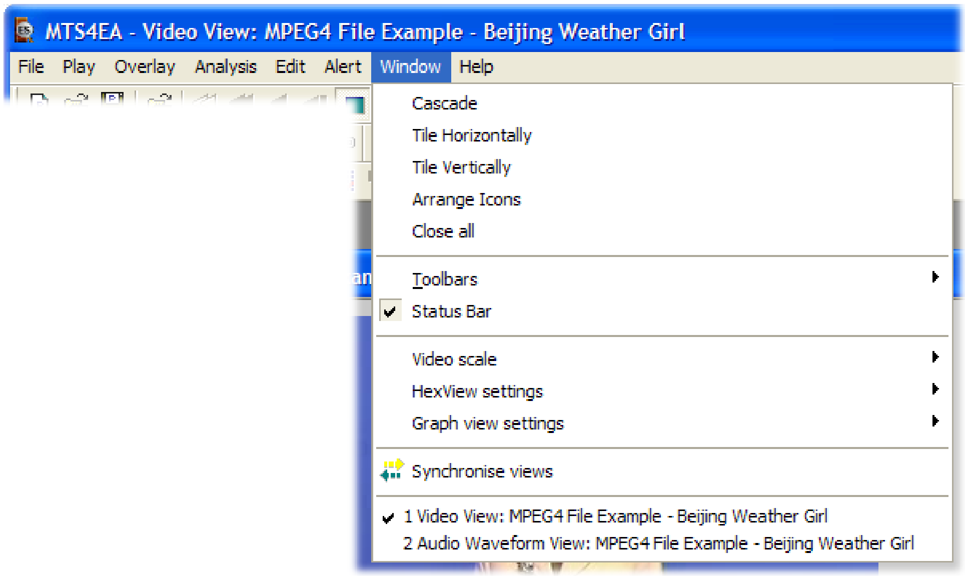
Overall alert level: All disabled - except one/more specified Warning alerts



NOTE: one/more Alerts disabled | All | Disabled: 5 | 0001

Overall alert level: All disabled - except one/more specific Error alerts

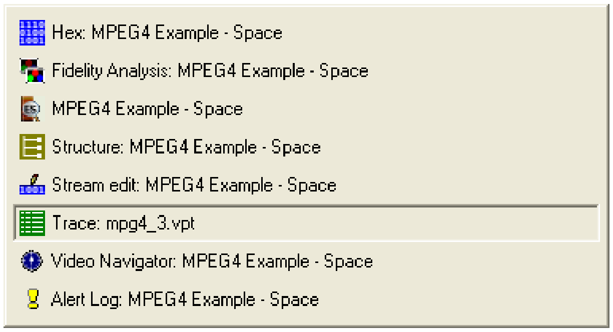
Window Menu



The commands on this menu are typical of a standard Windows application, such as arranging windows; list of open windows.

Quickly Switch Windows

Pressing `Ctrl+Tab` together displays a list of open MTS4EA windows:

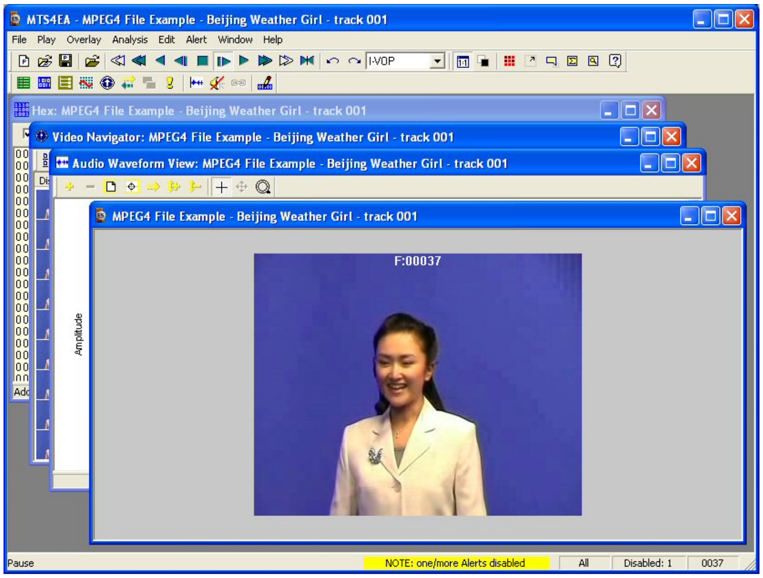

Hex: MPEG4 Example - Space
Fidelity Analysis: MPEG4 Example - Space
MPEG4 Example - Space
Structure: MPEG4 Example - Space
Stream edit: MPEG4 Example - Space
Trace: mpg4_3.vpt
Video Navigator: MPEG4 Example - Space
Alert Log: MPEG4 Example - Space

Repeatedly pressing `Ctrl+Tab` steps through all the open windows. Alternatively, the mouse can be used to select a window.

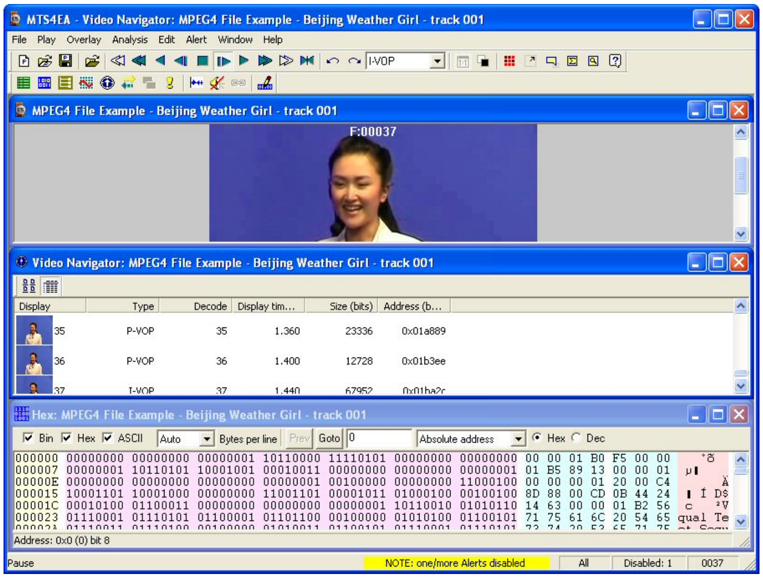
Cascade, Tile, Arrange Icons, Close All

These function as per standard Windows commands:

- Cascade arranges the open windows in a cascade:


MTS4EA - MPEG4 File Example - Beijing Weather Girl - track 001
File Play Overlay Analysis Edit Alert Window Help
I-VOP
Hex: MPEG4 File Example - Beijing Weather Girl - track 001
Video Navigator: MPEG4 File Example - Beijing Weather Girl - track 001
Audio Waveform View: MPEG4 File Example - Beijing Weather Girl - track 001
Amplitude
MPEG4 File Example - Beijing Weather Girl - track 001
F:00037
Pause
NOTE: one/more Alerts disabled
All
Disabled: 1
0037

- Tile Horizontally arranges the open windows as shown:

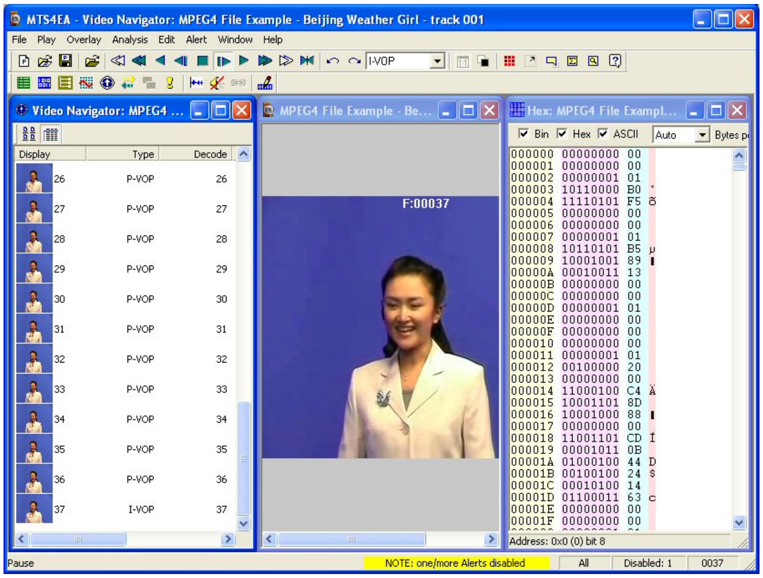

MTS4EA - Video Navigator: MPEG4 File Example - Beijing Weather Girl - track 001
File Play Overlay Analysis Edit Alert Window Help
I-VOP
MPEG4 File Example - Beijing Weather Girl - track 001
F:00037
Video Navigator: MPEG4 File Example - Beijing Weather Girl - track 001
Hex: MPEG4 File Example - Beijing Weather Girl - track 001
☑ Bin ☑ Hex ☑ ASCII
Auto
Bytes per line
Prev
Goto
0
Absolute address
● Hex ○ Dec
Pause
NOTE: one/more Alerts disabled
All
Disabled: 1
0037

Display	Type	Decode	Display tim...	Size (bits)	Address (b...
35	P-VOP	35	1.360	23336	0x01a889
36	P-VOP	36	1.400	12728	0x01b3ee
37	I-VOP	37	1.440	67952	0x01ba2c

```
000000 00000000 00000000 00000001 10110000 11110101 00000000 00000000 00 00 01 B0 F5 00 00   °õ
000007 00000001 10110101 10001001 00010011 00000000 00000000 00000001 01 B5 89 13 00 00 01  µ▌
00000E 00000000 00000000 00000000 00000001 00100000 00000000 11000100 00 00 00 01 20 00 C4        Ä
000015 10001101 10001000 00000000 11001101 00001011 01000100 00100100 8D 88 00 CD 0B 44 24  ▌ Í D$
00001C 00010100 01100011 00000000 00000000 00000001 10110010 01010110 14 63 00 00 01 B2 56  c   ²V
000023 01110001 01110101 01100001 01101100 00100000 01010100 01100101 71 75 61 6C 20 54 65 qual Te
```

Address: 0x0 (0) bit 8

- Tile Vertically arranges the open windows as shown:


MTS4EA - Video Navigator: MPEG4 File Example - Beijing Weather Girl - track 001
File
Play
Overlay
Analysis
Edit
Alert
Window
Help
I-VOP
Video Navigator: MPEG4 ...
MPEG4 File Example - Be...
Hex: MPEG4 File Exampl...
F:00037
☑ Bin
☑ Hex
☑ ASCII
Auto
Bytes p
Pause
NOTE: one/more Alerts disabled
All
Disabled: 1
0037

Display	Type	Decode
26	P-VOP	26
27	P-VOP	27
28	P-VOP	28
29	P-VOP	29
30	P-VOP	30
31	P-VOP	31
32	P-VOP	32
33	P-VOP	33
34	P-VOP	34
35	P-VOP	35
36	P-VOP	36
37	I-VOP	37

```
000000 00000000 00
000001 00000000 00
000002 00000001 01
000003 10110000 B0 °
000004 11110101 F5 õ
000005 00000000 00
000006 00000000 00
000007 00000001 01
000008 10110101 B5 µ
000009 10001001 89 ▮
00000A 00010011 13
00000B 00000000 00
00000C 00000000 00
00000D 00000001 01
00000E 00000000 00
00000F 00000000 00
000010 00000000 00
000011 00000001 01
000012 00100000 20
000013 00000000 00
000014 11000100 C4 Ä
000015 10001101 8D
000016 10001000 88 ▮
000017 00000000 00
000018 11001101 CD Í
000019 00001011 0B
00001A 01000100 44 D
00001B 00100100 24 $
00001C 00010100 14
00001D 01100011 63 c
00001E 00000000 00
00001F 00000000 00
```

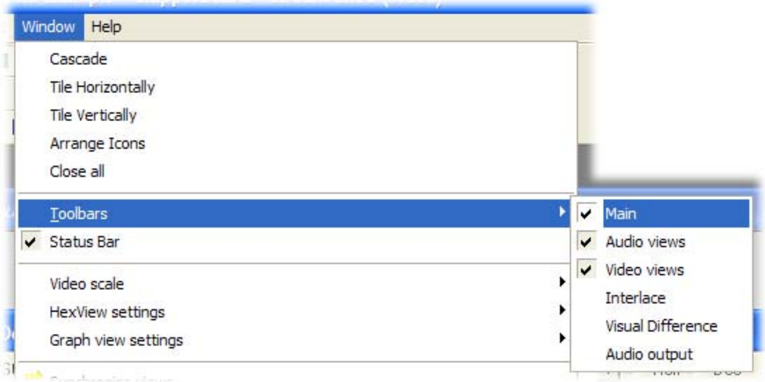
Address: 0x0 (0) bit 8

- Arrange Icons lines up the windows when they are minimized

- Close all closes all windows in the MTS4EA

Toolbars

In the following example, the Main and Views toolbars are shown, but not the Interlace and Visual Difference toolbars. When selected, a toolbar is displayed with the icons visible. This is indicated by the check mark in the Window menu. Toolbar buttons may be grayed out when a function is inactive.



Main

See *Functions of the Main Toolbar Buttons* on page 6–273 for more information.

Video Views

See *Functions of the Views Toolbar Buttons* on page 6–274 for more information.

Interlace

The Interlace toolbar is not shown by default; it is always turned on when a stream that is interlaced (or could be interlaced - in H.264/AVC this may not be known until later in the stream) is opened. See *Interlace Toolbar* on page 6–275 for more information.

Visual Difference

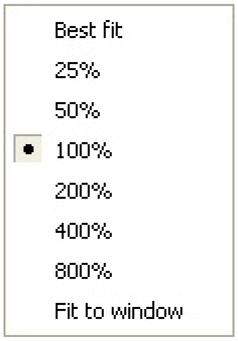
The Visual Difference toolbar is not shown by default; it is not shown until Visual Difference is enabled on the Overlay menu. See *Overlay Color* ◩ *`Ctrl+W`*, page 6–113 for more information.

Status Bar

When selected (which is the default setting), the Status Bar is displayed at the bottom of the window. When not selected, it is hidden.

See *Status Bar* on page 6–278 for more information.

Video Scale



This sets the size that the decoded video appears in the video window.

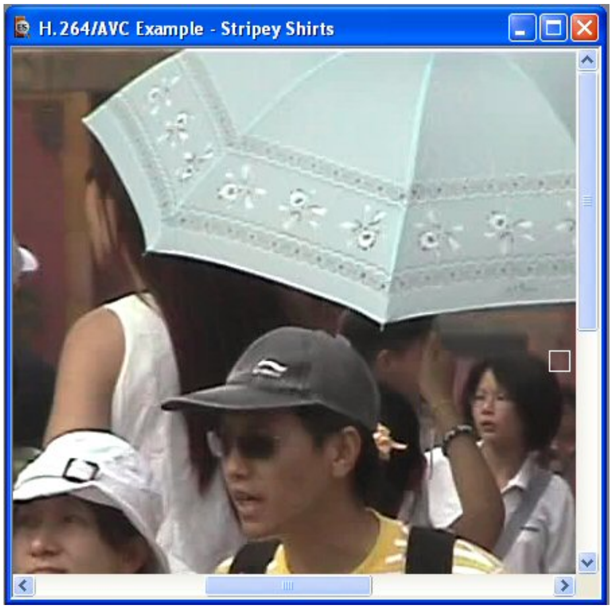
Best Fit

This zooms the displayed video as far as is possible within the video window, while still maintaining the original aspect ratio of decoded video.

50%; 100% (1:1); 200%; 400%; 800%

These set the displayed video to the selected size in the video window.

If the selected size is larger than the available window area, then a section of the video window will be displayed, with scroll bars:

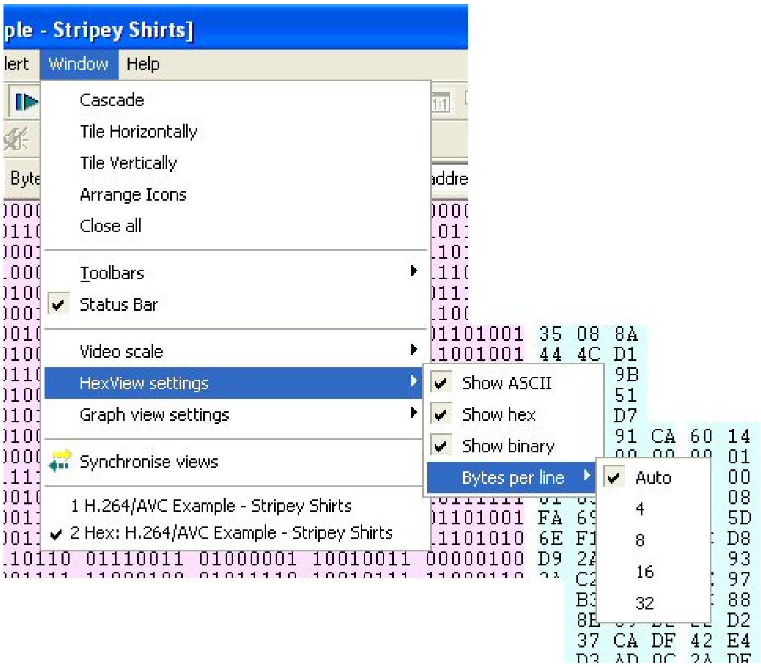


When the MTS4EA starts, it is set by default to 100% or 1:1; the video displays at its actual size with one screen pixel equaling one pixel in the displayed video.

Fit to Window

This zooms the displayed video to completely fill the video window; it does not maintain the original aspect ratio of decoded video.

HexView Settings



The HexView menu has four options:

- Show ASCII
- Show hex
- Show binary
- Bytes per line

Show ASCII, Show hex, Show binary

These options individually turn on/off the display of the specified area of the HexView.

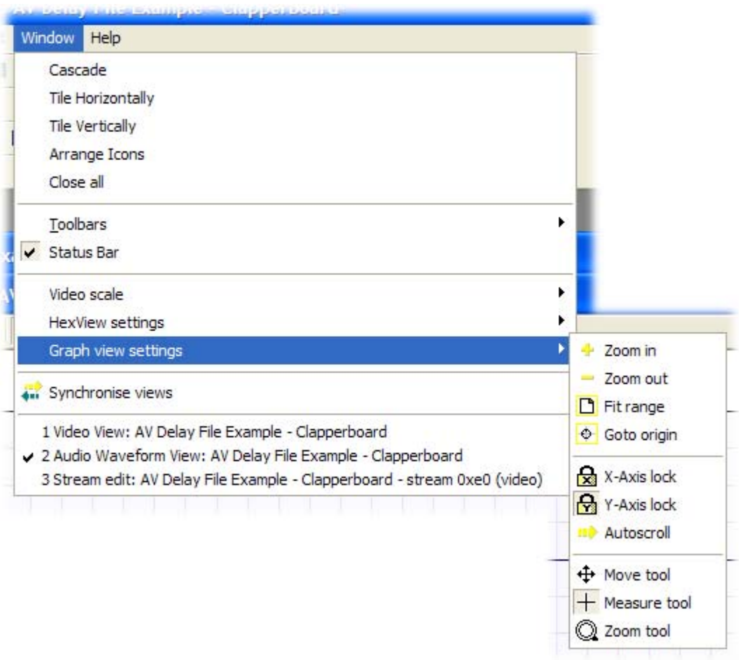
They can be turned on/off from either this menu or from the top line of the HexView window itself, when it is open.

Bytes per line

This sets the displayed number of bytes per line:

- Auto sets the maximum number of bytes that will fit within the active HexView window at the size it is, allowing for the visible ASCII/hex/binary areas.
- When 4, 8, 16 or 32 is selected, then the given number of bytes are displayed, and if the display is too wide for the active HexView window, a horizontal scroll bar is displayed along the lower edge of the HexView window.

Graph View Settings



The Graph view menu has ten options, which control how the data appears on the Buffer analysis graph. These options are described in the following table:

Icon	Equivalent toolbar icon	Function
+ Zoom-in	+	Zoom in (centered on current window)
− Zoom-out	−	Zoom out (centered on current window)
□ Fit to range	□	Fit all data into the visible windo
⊙ Goto origin	⊙	Go to the origin (time = 0)
🔒 X-Axis lock	🔒	Lock the X-axis when zooming/scrolling
🔒 Y-Axis lock	🔒	Lock the Y-axis when zooming/scrolling
⇨ Autoscroll	⇨	Autoscroll to follow frames as decoded
✥ Move tool	✥	Move window left/right/up/down
+ Measure tool	+	Measure the values at center point of +
🔍 Zoom tool	🔍	Zoom in/out, centered on cursor
Goto view ▸	n/a	See *Synchronized Views/Navigating the Views* on page 6–7
Synchronise views	n/a	

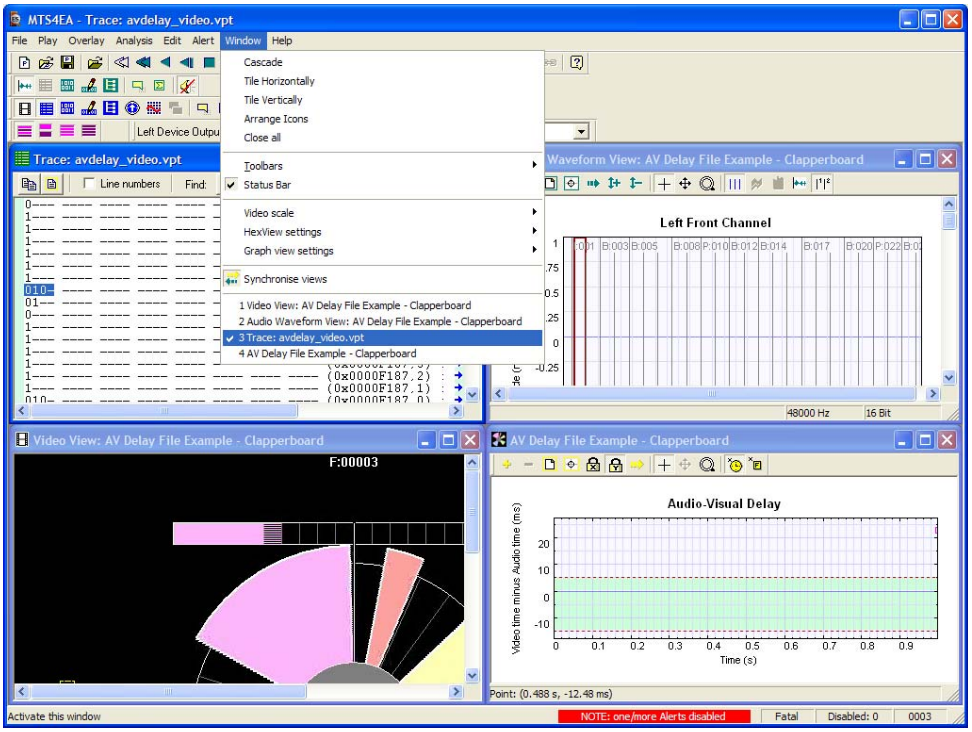
Synchronize views

When Window > Synchronise Views ( ) is selected, all open windows synchronize automatically.

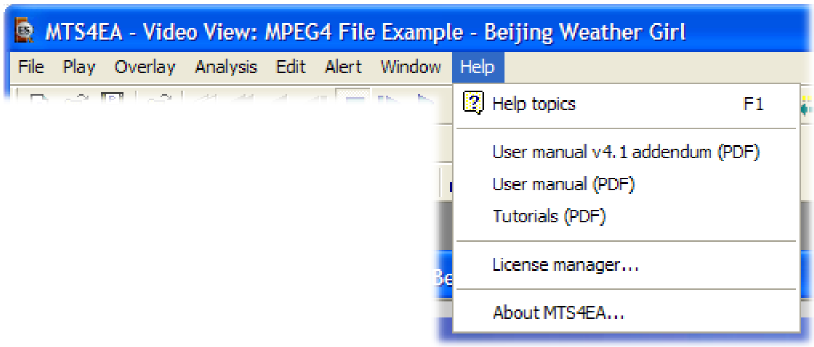
For example, if the Video navigator and Hex views are both open at the same time as the Audio waveform view, the Synchronize views icon is pushed in and the stream is played. The individual video frames appear in the navigator view and the highlighting in the HexView window advances so that all views display the same relative point in the stream at any one time.

Active Views

A list of all of the open views is displayed at the bottom of the Windows menu. A check mark appears next to the currently selected view. Highlight an entry to change the selection.


MTS4EA - Trace: avdelay_video.vpt
File
Play
Overlay
Analysis
Edit
Alert
Window
Help
Cascade
Tile Horizontally
Tile Vertically
Arrange Icons
Close all
Toolbars
Status Bar
Video scale
HexView settings
Graph view settings
Synchronise views
1 Video View: AV Delay File Example - Clapperboard
2 Audio Waveform View: AV Delay File Example - Clapperboard
3 Trace: avdelay_video.vpt
4 AV Delay File Example - Clapperboard
Left Device Outpu
Trace: avdelay_video.vpt
Line numbers
Find:
Waveform View: AV Delay File Example - Clapperboard
Left Front Channel
48000 Hz
16 Bit
Video View: AV Delay File Example - Clapperboard
F:00003
AV Delay File Example - Clapperboard
Audio-Visual Delay
Video time minus Audio time (ms)
Time (s)
Point: (0.488 s, -12.48 ms)
Activate this window
NOTE: one/more Alerts disabled
Fatal
Disabled: 0
0003

Help Menu


MTS4EA - Video View: MPEG4 File Example - Beijing Weather Girl
File Play Overlay Analysis Edit Alert Window Help
Help topics F1
User manual v4.1 addendum (PDF)
User manual (PDF)
Tutorials (PDF)
License manager...
About MTS4EA...

This menu provides access to the Help information (on-line, PDF), the tutorials, configuration information, and license information (including licensed options).

Help topics

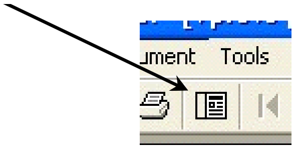
The help is organized into topics and is displayed when the F1 key is pressed or ? icon on the toolbar is clicked.

User manual (PDF)

This displays a PDF version of this user manual. This is provided because:

- It is sometimes preferable to use this as a source of Help
- There is a full table of contents, allowing easy navigation through the whole manual
- There are thumbnails of all pages, which can be useful for navigation through the manual
- It is easier to print out complete sections of the manual, or further copies of the complete manual

To enable the navigation items in Adobe Acrobat, click the Navigation pane icon:



Then click the Bookmarks tab to see the list of topics.

See the Adobe Acrobat documentation for details about how to navigate through these views.

Tutorials (PDF)

This displays a PDF version of the tutorials on how to optimize your use of the MTS4EA. It is strongly recommended that you work through the tutorials. The tutorials are also included in the standard Help (accessed via the F1 key), although the PDF version is easier to print out than the copy included in the standard help system.

License manager...

This displays the MTS4EA License manager, including:

- Current license status information
- Other possible MTS4EA options that can be licensed
- Option to update the license key

See chapter 4 on *Installation and Licensing* for more information on the license manager.

About MTS4EA...

This displays the exact version number of the MTS4EA and the copyright message, and allows access to the MTS4EA build options.

Icon Toolbars

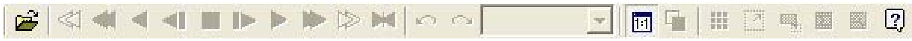
These toolbars provide quick selection of some of the commonly used functions. There are two main icon toolbars, which can be individually moved and docked:

- Main toolbar, containing functions such as play the video and open files.
- Views toolbar, to access the views other than the video views.

NOTE. *There are other toolbars for individual windows and controls, for example for Interlace and Buffer analysis. This section refers to only the main icon toolbars.*

Disabled Toolbar Buttons

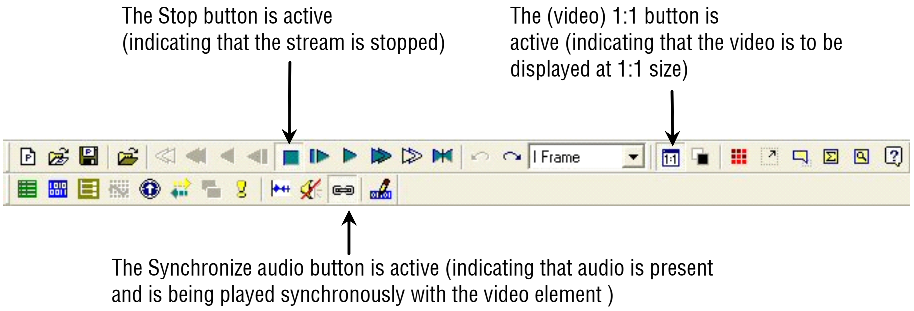
At various times, some of the toolbar buttons are grayed out, indicating that they are disabled because that function cannot be active at the time.



The only buttons available in the above example are the Open video... and Help buttons.

Active/inactive Toolbar Buttons

When a toolbar icon looks as though it is pressed in, it is active (selected).



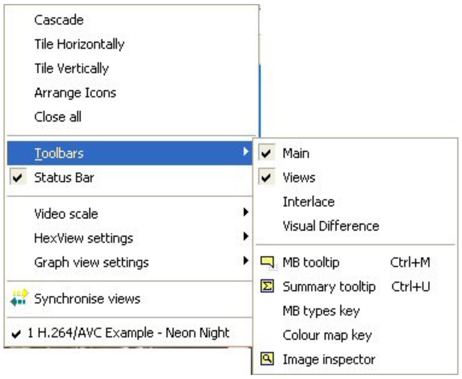
The Stop button is active (indicating that the stream is stopped)

The (video) 1:1 button is active (indicating that the video is to be displayed at 1:1 size)

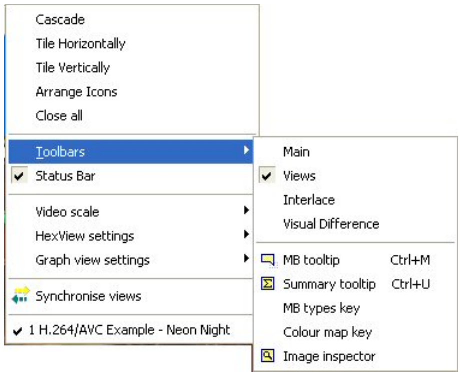
The Synchronize audio button is active (indicating that audio is present and is being played synchronously with the video element)

Displaying/Hiding the Main Toolbar

The Main toolbar can be displayed or hidden using the check box on the Toolbars submenu of the Window menu:



Toolbar displayed



Toolbar hidden

Moving/Docking

The Main toolbar is docked at the top of the window below the main menu by default, but it can be dragged from this position to be floating or be docked on any of the other three edges of the main window, by clicking and dragging the narrow vertical line at the left edge of the toolbar:

The Views toolbar is normally docked below the Main toolbar, but can be moved/undocked as required.

Functions of the Main Toolbar Buttons

Each of the Main toolbar buttons provides quick access to a specific function of the MTS4EA, as follows:

Toolbar icon	Ctrl +	Name	Description	Page
📂	O	Open stream…	Open a video/audio file	6–4 and 6–16
▶	P	Play forward	Play a video/audio file (forwards)	6–51
⏩	F	Fast forward	Fast forward a video file	6–53
▷▷	B	Blind fast forward	Blind fast forward a video file (video blanked)	6–54
❙▶	A	Pause/Step forward	Pause a video file/advance by one frame	6–55
■	S	Stop	Stop playing a video/audio file	6–54
▶◀		Pause on frame…	Pause on specific frame number	6–59
◀	Shift + P	Play backward	Play a video file backwards	6–55
⏪	Shift + F	Fast backward	Fast backward a video file	6–56
◁◁	Shift + B	Blind fast backward	Blind fast backward a video file (video blanked)	6–56
◀❙	Shift + A	Pause/Step backward	Pause a video file/ step back by one frame	6–56
↷	K	Skip forward	Skip to next I-frame/forward n frames/forward n seconds	6–56
↶	Shift + K	Skip backward	Skip to previous I-frame/backward n frames/backward n seconds	6–57
1:1	1	1:1	Set video scale: Best fit or 1:1	6–263
◧	W	Black/White	Set overlay digits to black/white	6–113
▦	Y	MB types	MacroBlock type overlay	6–94
↗	E	Motion vectors	Overlay motion vectors display	6–98
🗨	M	MB tooltip	Open/close the MacroBlock tooltip	6–84
Σ	U	Summary tooltip	Open/close the Summary tooltip	6–74
🔍	I	Image inspector	Open/close the Image inspector	6–92
?	F1	Help *	Go to Help topics	6–269

* The F1 key is pressed without pressing the Ctrl key

Functions of the Views Toolbar Buttons

Each of the Views toolbar buttons provides quick access to a specific function of MTS4EA.

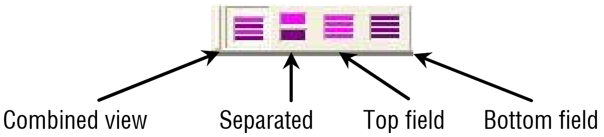
Toolbar icon	Ctrl +	Name	Description	Page
▦	Shift+T	View trace…	View the currently selected Trace file	6–168
1010 0011	H	View stream hex…	Open the current steam in the HexView	6–176
▤	R	View file structure…	Open the current video/container file and view the structure in navigable tree form	6–184
📈		View buffer analysis…	Graphs of VBV/VCV/VMV. (depends upon standard)	6–186
⊕		Video navigator…	Show thumbnail viewer of each video frame and basic information	6–168
⇄		Synchronize views	Synchronize all open views	6–7
◧		View fidelity analysis…	Show fidelity analysis	6–196
📊		View graphical analysisi	Show graphical analysis (in Excel)	
💡		View alert log…	Display log of alerts	6–251
⇹		Audio waveform view…	Open the current audio stream on the Audio waveform view	6–124
🔇		Mute audio	Turn audio off/on	6–69
⛓		Synchronize audio	Synchronize audio and video streams	6–69
✎		Edit video stream…	Open the stream for editing	6–231

Context-sensitive Toolbars/Tooltips

These are toolbars and tooltips (or sometimes information windows) that only appear or are only available (not grayed out) for specific video standards or specific functions.

NOTE. *These toolbars dock and are minimized/maximized in the same way as standard Windows toolbars, and in the same manner as the MacroBlock and Summary tooltips.*

Interlace Toolbar



Only one of these four buttons is pushed in (active) at any time.

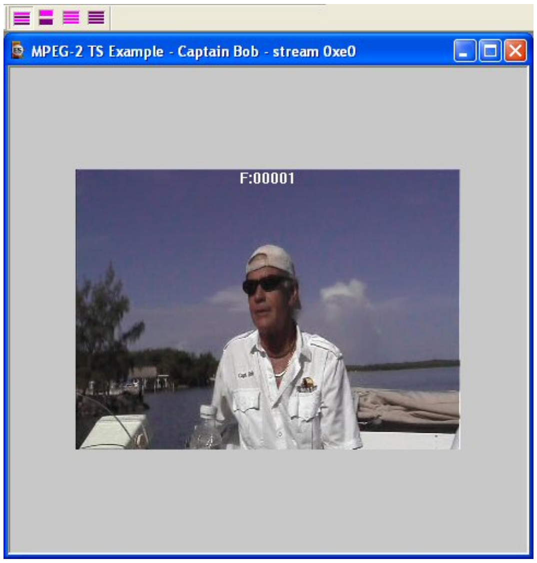
NOTE. *The Interlace toolbar only appears when a bitstream could have Interlace within it. The fact that the Interlace toolbar is there does not necessarily mean that the bitstream is interlaced – only that interlace is permissible/possible in the given standard, Profile and Level.*

These are examples of video streams where the Interlace toolbar can appear (this is not a complete list):

- H.264/AVC (All supported profiles)
- VC-1 Advanced profile
- MPEG-4 Advanced simple profile, levels 4 and 5
- MPEG-2 Main profile, Main level

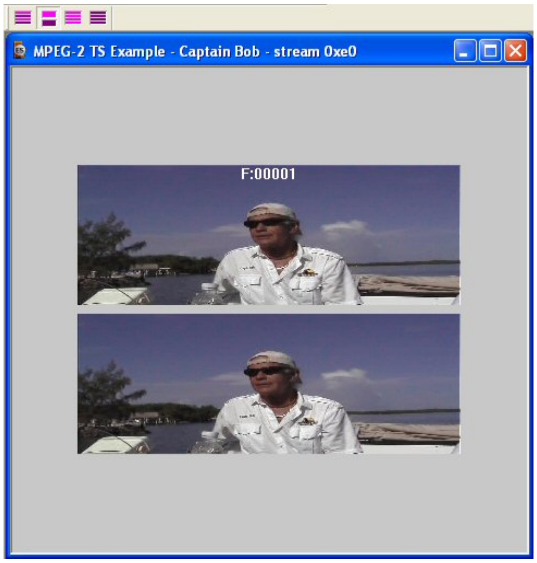
NOTE. *When the Interlace toolbar has not been automatically displayed by the MTS4EA (or it has been closed), it can be displayed by clicking the Windows menu, and then selecting Toolbars, Interlace.*

Combined frame view. Both fields are shown together, as a single image:



NOTE. *Some data and functions are not available when the video is displayed in this view. For example, the MacroBlock tooltip is empty and cannot be used, and motion vectors, MacroBlock Types and other overlaid data do not appear.*

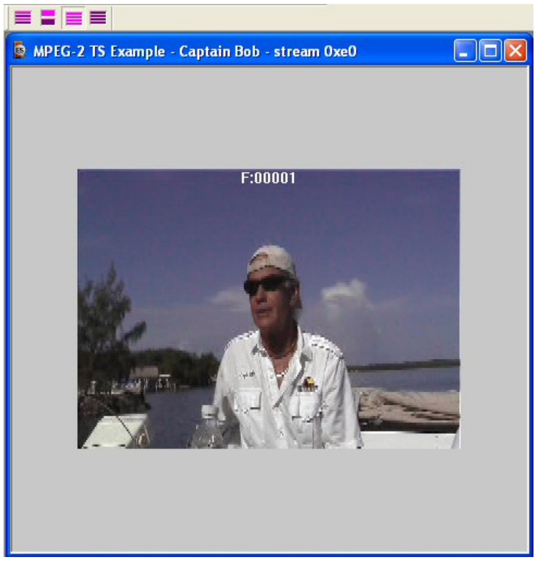
Separated fields view. The top and bottom fields are shown separately, one above the other:



Top-field view. Only the top-field image data is displayed. For field coded MacroBlocks, the MB tooltip and overlays show data relevant to the top field.

For frame coded MacroBlocks, the MB tooltip and overlays show data relevant to the frame.

Each row is shown twice (duplicated) to produce an image to the full image height:



Bottom-field view. As above but for bottom field.

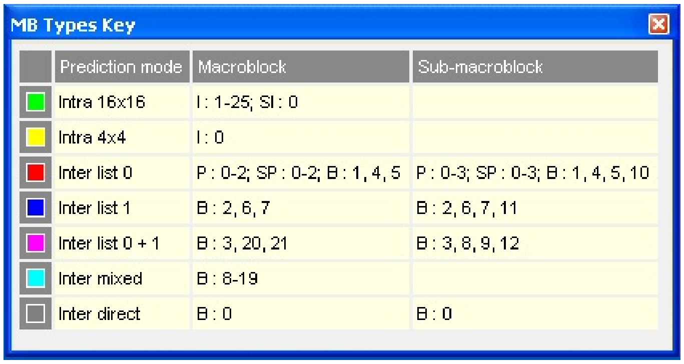
MacroBlock Types Color Key Tooltip

This is an information tooltip that appears when the MacroBlock Types overlay is on: its function is to explain the colors used. This appears when the MB types overlay is on, the ▦ icon is active.

NOTE. *To force undocking of the MB types color key tooltip, hold the Ctrl key while dragging with the mouse.*

The MB types color key tooltip can also be turned on again by going to the Window menu, selecting Toolbars, and then clicking the MB types key.

This tooltip can be closed by clicking on the X at the top of the tooltip; to redisplay it click the MB types overlay icon off then on.

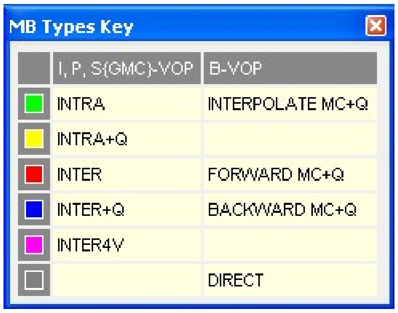


MB Types Key

	Prediction mode	Macroblock	Sub-macroblock
🟩	Intra 16x16	I : 1-25; SI : 0	
🟨	Intra 4x4	I : 0	
🟥	Inter list 0	P : 0-2; SP : 0-2; B : 1, 4, 5	P : 0-3; SP : 0-3; B : 1, 4, 5, 10
🟦	Inter list 1	B : 2, 6, 7	B : 2, 6, 7, 11
🟪	Inter list 0 + 1	B : 3, 20, 21	B : 3, 8, 9, 12
■	Inter mixed	B : 8-19	
⬜	Inter direct	B : 0	B : 0

H.264/AVC example

NOTE. *The Intra inferred MacroBlock type was shown in yellow in previous versions of MTS4EA. However, because it occurs infrequently and is a 16x16 type, it is now grouped in the Intra 16x16 MacroBlock type, and the Intra 4x4 MacroBlock type is now shown in yellow.*



MB Types Key

	I, P, S(GMC)-VOP	B-VOP
🟩	INTRA	INTERPOLATE MC+Q
🟨	INTRA+Q	
🟥	INTER	FORWARD MC+Q
🟦	INTER+Q	BACKWARD MC+Q
🟪	INTER4V	
⬜		DIRECT

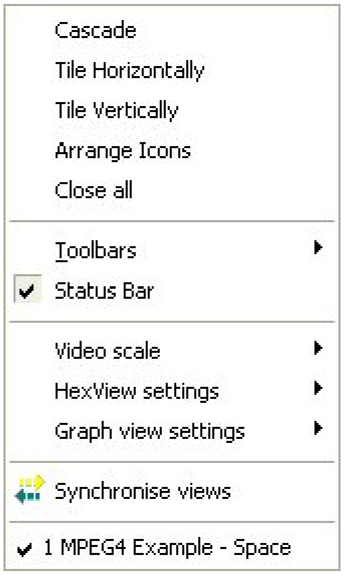
MPEG-4 example

Status Bar

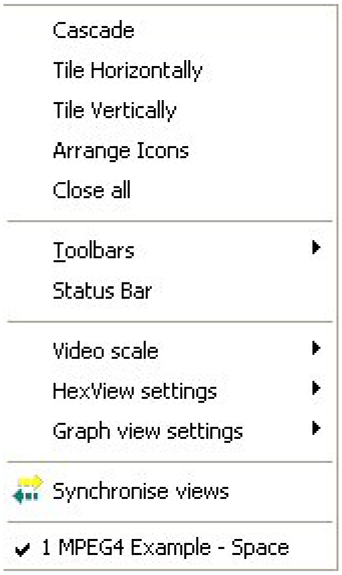
The status bar provides:

- A one-sentence description of each menu item, as the mouse is moved over the menu item
- The stream Play mode (see *Playing Mode: Restrictions* on page 6–6)
- Context-sensitive information. The information displayed depends on which window type is open and the current status of the file within that window

The status bar can be displayed or hidden from within the Window menu, in the same way as the icons toolbar:

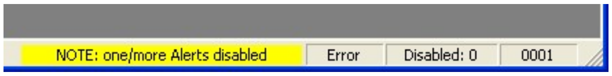
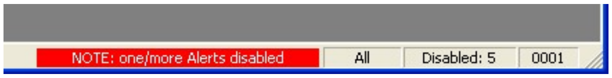
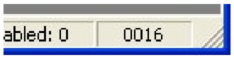
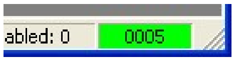
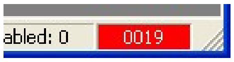
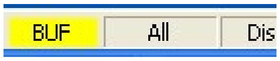


Status bar displayed



Status bar hidden

For more information on status bar indications of:

Alert levels, alert warnings, alerts disabled:  	See page 6–258
Frame range (in/out):   	See page 6–112
When custom parameters are used for buffer analysis, this is indicated in the status bar of the main MTS4EA window by the word BUF: 	See pages 6–191 and 6–192

Ctrl Shortcut Keys

All shortcut keys are accessed by holding the Ctrl key and pressing the letter given in the following table, except the F1 key:

Toolbar icon	Ctrl +	Name	Description	Page
	mouse	Force undock	Force undocking of tooltip (for example, Summary/MacroBlock)	-
1:1	1	1:1	Set video scale: Best fit or 1:1	6–263
❙▶	A	Pause/Step forward	Pause a video file/advance by one frame	6–55
◀❙	Shift + A	Pause/Step backward	Pause a video file/ step back by one frame	6–56
▷▷	B	Blind fast forward	Blind fast forward a video file (video blanked)	6–54
◁◁	Shift + B	Blind fast backward	Blind fast backward a video file (video blanked)	6–56
⎘	C	Copy	Copy highlighted selection	6–233 and 6–243
↗	E	Motion vectors	Overlay motion vectors display	6–98
▶▶	F	Fast forward	Fast forward a video file	6–53
◀◀	Shift + F	Fast backward	Fast backward a video file	6–56
	G	Graph enable	Enable the graph output	6–153
1010 0011	H	View stream hex…	Open the current steam in the HexView	6–176
🔍	I	Image inspector	Open/close the Image inspector	6–92
↷	K	Skip forward	Skip to next I-frame/forward n frames/forward n seconds	6–56
↶	Shift + K	Skip backward	Skip to previous I-frame/backward n frames/backward n seconds	6–57
🗨	M	MB tooltip	Open/close the MacroBlock tooltip	6–84
📂	O	Open stream…	Open a video/audio file	6–4 and 6–16
▶	P	Play forward	Play a video/audio file (forwards)	6–51
◀	Shift + P	Play backward	Play a video file backwards	6–55
☰	R	View file structure…	Open the current video/container file and view the structure in navigable tree form	6–184
■	S	Stop	Stop playing a video/audio file	6–54
	T	Trace enable	Enable the Trace output	6–136

Toolbar icon	Ctrl +	Name	Description	Page
▤	Shift+T	View trace…	View the currently selected Trace file	6–168
Σ	U	Summary tooltip	Open/close the Summary tooltip	6–74
📋	V	Paste	Paste cut/copied selection in Bitstream editor view	6–233
◪	W	Black/White	Set overlay digits to black/white	6–113
✂	X	Cut	Cut highlighted selection in Bitstream editor view	6–233
▦	Y	MB types	MacroBlock type overlay	6–94
↶	Z	Undo last edit	Undo last edit in Bitstream editor view	6–233
	Tab	Switch windows	Quickly switch between open windows	6–259
?	F1	Help *	Go to Help topics	6–269
	F3	Find next *	In Trace and HexView only	6–182
	Shift+F3	Find previous *	In Trace and HexView only	6–182

* The F1, F2 & F3 keys are pressed without pressing the Ctrl key

Alt Menu Keys

Menus can be selected by pressing the Alt key and the letter underlined in the menu option name. For example, pressing Alt+f will open the File menu. The cursor keys (or the mouse) can be used to select the option required.

Command Line/Batch Mode

This feature allows the MTS4EA to be run in one of two ways:

- Command line mode
- Batch mode

Command line mode

In command line mode, the MTS4EA opens with the Windows display as usual, but using the files and flags specified in the command line.

Batch mode

In batch mode, there is no Windows display (the window is actually minimized and not activated) and the input file, flags and output files are specified in the command line.

The exception to this in batch mode is when no input file is specified. In this case, the MTS4EA opens in Windows mode, allows you to specify a bitstream file name, and then closes Windows mode and goes back to batch mode.

The MTS4EA exits automatically from batch mode when the last selected frame has been decoded.

NOTE. *Remember the –b option to run MTS4EA in batch mode (rather than command-line mode).*

Running the MTS4EA in command line/batch mode

To use command line or batch mode:

- Use Windows Start/Run and type in the line
- Open a command prompt window (typically from Windows Start/Programs/Accessories) and enter the line

The format of the line to be entered is:

`MTS4EA [options...] <filename>` where:

- `[options...]` is zero or more of the options listed in 0

- `<filename>` is the input video file to be decoded

Specification of Filename (input video file)

The input video filename must be the full filename (including any file extension after the . [period]).

Using demonstration sequences in batch mode. To use batch mode with the demonstration tutorial video sequences provided on the File menu, the filename for these sequences is specified below:

Demo sequence	Standard	Filename
H.264/AVC byte streams		
Neon Night	H.264/AVC Baseline Profile/Level 2	avc_1
Canary Wharf	H.264/AVC Extended Profile/Level 3	avc_2
Bus Junction	H.264/AVC Baseline Profile/Level 3	avc_3
Grenadier Guards	H.264/AVC Main Profile/Level 3	avc_4
Stripey Shirts	H.264/AVC High Profile/10	avc_5
Times Square	H.264/AVC High Profile/4:4:4	avc_6
VC-1 Advanced stream		
Central Park	VC-1 Advanced	wmv9_1
MPEG-4 Elementary Streams		
Woman Drinking	MPEG-4 Main Profile	mpg4_1
Train in Station	MPEG-4 Main Profile	mpg4_2
Space	MPEG-4 Simple Profile	mpg4_3
Man Walking	MPEG-4 Advanced Simple Profile	mp4asp_1
Synthetic	MPEG-4 Advanced Simple Profile	mp4asp_2
Window Car	MPEG-4 Advanced Simple Profile	mp4asp_3
H.263 streams		
Rally (250k)	H.263 Baseline/MPEG-4 Short Header	h263_1
Rally (2M)	H.263+ with Annex T	h263_2
Person Track	H.263+ with Annexes D, F, I, J, S, T	h263_3
MP4 files		
Packet Woman	MP4 file containing MPEG-4 Simple Profile/ Level 1	mp4fil_1
Piccadilly Circus	MP4 file containing MPEG-4 Simple Profile/ Level 2	mp4fil_2
Beijing Weather Girl	MP4 file containing MPEG-4 Simple Profile/ Level 5 (with audio)	mp4fil_3

Demo sequence	Standard	Filename
Las Vegas	H.264/AVC Baseline Profile/Level 1.2	mp4fil_4
Live Band	MPEG-4 Simple Profile/Level 1 (video); AAC HE (audio)	audio_4
Stage Interview	MPEG-4 Simple Profile/Level 1 (video); AAC Main (audio)	audio_5
3GPP file		
Mobile Hands	3GPP file containing MPEG-4 Simple Profile/ Level 1	3gpfil_1
MPEG-2 Program Streams		
Bus Junction	MPEG-2 Main Profile/ Main Level	mp2_1
Person Track	MPEG-2 Main Profile/ Main Level	mp2_2
Grenadier Guards	MPEG-2 Main Profile/ Main Level	mp2_3
MPEG-2 Transport Streams		
Golden Gate	H.264/AVC Main Profile	mp2ts_1
Mangroves	MPEG-2 MP@ML	mp2ts_2
Captain Bob	MPEG-2 MP@ML (with MPEG-1 Audio Layer II)	mp2ts_3
British Grand Prix		
Microsoft® ASF files		
Beach Girl	Simple Profile	asf_1
Great Wall	Main Profile	asf_2
ADTS Streams		
Dance Track	AAC LC	audio_6
A/V Delay Stream		
Clapperboard	MPEG-2 Main Profile / Main Level (video) Dolby AC-3 (audio)	avdelay

To use these demonstration sequences, the option `-d` is used: see *List of options* on page 6–286.

In addition, the following YUV files are provided for fidelity analysis and visual differencing with the example streams:

YUV files	Use with compressed demo sequence	Frame rate	Number of frames
`guards_yuv`	H.264/AVC Grenadier Guards MPEG-2 Grenadier Guards	25	10
`man_walking_yuv`	MPEG-4 Man Walking	30	All

Command Line/Batch Mode Options

Form of options. All options take the following form:

```
-x <value>
```

(Where `x` is the option and `<value>` is the value entered; for some options there is no value.)

NOTE. *All options must be entered in lower case.*

If there is a value, there is always a <space> between the `-x` and the value.

All options must be separated by spaces.

NOTE. *Where filenames or folders have spaces in the path or name, these must be put inside double quotes.*

There is limited checking on the options/option values.

List of options.

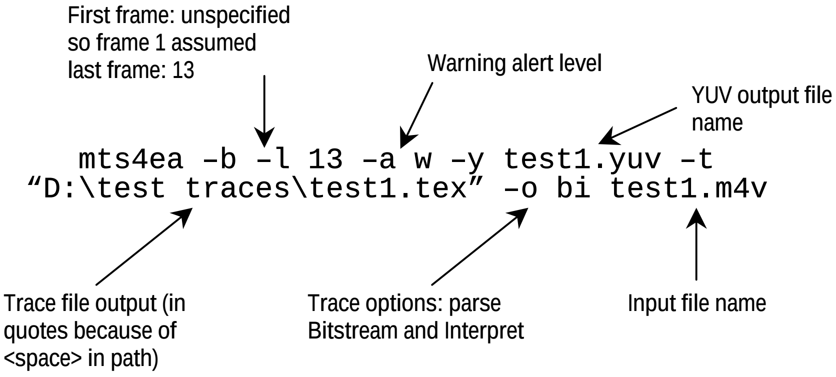
Option	Value (if any)	Description
-a	<alertlevel>	Alert level, which must be one of the following text strings: error warning fatal As with MTS4EA run in Windows mode, if the warning alert level is on, trace outputs will also include error and fatal level alerts. Fatal level alerts are always on.
-b		When present, enable batch mode. If not present, command line mode is used.
-d		When this flag is present, it signifies that one of the demonstration tutorial video sequences as provided with the MTS4EA is to be used as the input video file. The particular demonstration sequence to be used is specified by the filename as listed under *Using demonstration sequences in batch mode* on page 6–284. (This filename is given at the end of the command line, like any other input filename.)
-f	<firstframe>	Integer value specifying the first frame to be used for the YUV and Trace outputs. If this option is not specified then frame number 1 (the first frame in the video sequence) is used.

Option	Value (if any)	Description
-h		Displays help Window (MTS4EA opens after OK is clicked).
-i	<trackID>	Where the input file is a container file such as MP4 or 3GPP or MPEG-2 Program Stream, there can be more than one video track in the file. If -i is not specified then the first video track found is used. If the track ID specified is not present, a file missing error is given.
-l	<lastframe>	Integer value specifying the last frame to be used for the YUV and Trace outputs. If this option is not specified, the last frame in the video sequence is used.
-m	<size>	Integer value specifying the maximum trace file size, in MB. If -m is not given or the value is set to 0, there is no limit on the size of the trace file (other than the available disk space).
-o	<options>	The Trace file options are any combination of the following letters with no spaces between them: Option Trace output generated f **F**rame summary g **G**OB summary m **M**acroBlock summary b parse **B**itstream d **D**CT level p **P**ixel level i **I**nterpret
-r	<yuvfile>	Full name of the YUV reference file used for fidelity analysis. (See also the options -s, -u and -v.)
-s	<size>	Header skip of the YUV reference file used for fidelity analysis. (See also the options –r, -u and -v.)
-t	<tracefile>	Full name of the Trace output file. All Trace outputs from the options are put in this file, in the order of the decoded bitstream (for example, the parse Bitstream and the Interpret trace outputs will be adjacent in the trace file for each MacroBlock). If no Trace output file is specified, the trace options are ignored.

Option	Value (if any)	Description
-u	<number>	Frame rate of the YUV reference file used for fidelity analysis. The number can be an integer, or a fraction expressed as a/b (where a and b are integers) or a decimal number. (See also the options –r, -s and -v.)
-v	<string>	Metric used for fidelity analysis: the string value must be one of the following: psnr255 PSNR with 255 signal range psnritu PSNR with ITU-R BT.601 signal range rmse Root Mean Square Error mse Mean Square Error mad Mean Absolute Differences sad Mean Absolute Differences (See also the options –r, -s and -u.)
-w	<error_file>	Output trace file with warnings/errors/fatal alerts only - no other trace information. <error_file> is the name of the file that holds the warnings/errors/fatal alerts. If there are no warnings/errors/fatal alerts, the <error_file> is zero length
-x	<options>	The format of image samples in uncompressed video files for input or output. This option is used when working with H.264/AVC High Profile to denote: The correct format for the uncompressed input when doing PSNR analysis The format for uncompressed output The format option is one of the following codes: 1 - one byte per sample 2le - two bytes per sample, little-endian 2be - two bytes per sample, big-endian
-y	<yuvfile>	Full name of the YUV output file (see *Format of Uncompressed Video File Output from Batch Mode* on page 6–289).

Example Command Line

An example of a valid command line is:



```
mts4ea –b –l 13 –a w –y test1.yuv –t
“D:\test traces\test1.tex” –o bi test1.m4v
```

Format of Uncompressed Video File Output from Batch Mode

NOTE. *The uncompressed video output file in this section is the uncompressed video output resulting from decoding the compressed video; this is a different file than the uncompressed video reference file that is used when doing fidelity analysis.*

The YUV data is either:

- 8 bits per sample, 4:2:0
- More than 8 bits per sample, and/or 4:2:2 or 4:4:4 (as used by H.264/AVC High Profile/FRExt, High/10, High/4:2:2, High/4:4:4)

YUV format o 8 bits per sample 4:2:0. The YUV file output is raw YUV with no headers of any kind. This is the same format as used by the Microsoft MPEG-4 Part 2 reference encoder Reference [7] (see *Compression Standards and File Types* section) and as used commonly by other programs:

- No headers of any kind (no file or frame headers)
- One byte per sample
- Row raster order (top picture row first)
- Planar YUV 4:2:0 sub-sampled (4 bytes of Y data for each byte of U data and each byte of Y data)

- Y plane values are 0-255 unsigned
- U and V plane values are unsigned with a DC offset of 128

Other uncompressed formats. The general uncompressed video file format is as follows:

- No headers of any kind (no file or frame headers)
- Concatenated planar image data
- Row raster order (top picture row first)
- Unsigned samples

For 8-bit sample depth:

- One byte per sample

For 9-16 bit sample depth:

- Two bytes per sample
- Both little- and big-endian byte orders supported

For YUV format:

- Concatenated Y, U and V planes
- U and V planes sub-sampled as required
- Y plane samples are unsigned
- U and V plane samples are unsigned with a DC offset of 2^{n-1}, where *n* is the chroma sample bit depth

For RGB format:

- Concatenated R, G and B planes

For grayscale format:

- Luma plane only

Bitstream Syntax Debugging

Bitstream Syntax Debugging

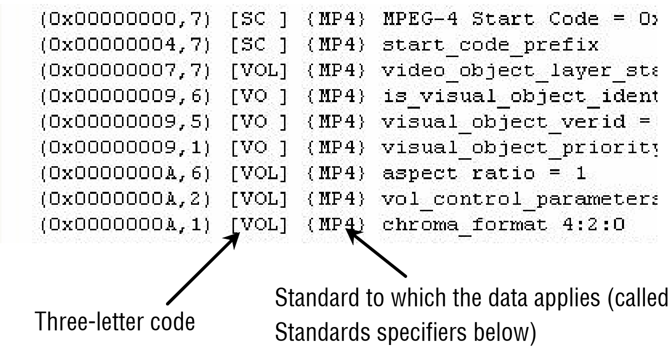
This section gives some tips and suggestions on debugging the syntax of video bitstreams syntax.

In order to understand these, you will need to have the relevant video standard(s) to hand.

NOTE. *This manual section applies to streams that only contain video, such as MPEG-4 Elementary Streams - it does not apply to debugging container files such as MP4/3GPP, which can also contain audio and other data.*

General codes used in Trace files and Alerts

Trace data and Alerts will often have two sets of information fields, such as shown below:



```
(0x00000000,7) [SC ] {MP4} MPEG-4 Start Code = 0x
(0x00000004,7) [SC ] {MP4} start_code_prefix
(0x00000007,7) [VOL] {MP4} video_object_layer_sta
(0x00000009,6) [VO ] {MP4} is_visual_object_ident
(0x00000009,5) [VO ] {MP4} visual_object_verid =
(0x00000009,1) [VO ] {MP4} visual_object_priority
(0x0000000A,6) [VOL] {MP4} aspect ratio = 1
(0x0000000A,2) [VOL] {MP4} vol_control_parameters
(0x0000000A,1) [VOL] {MP4} chroma_format 4:2:0
```

Three-letter code
Standard to which the data applies (called Standards specifiers below)

Explanation of Three-Letter Codes

Three-letter codes are used within the outputs generated by Trace/Interpret, Alerts and (sometimes) Trace/Parse Bitstream.

The codes are used to indicate the part of the picture hierarchy that applies to the text data given.

Three-letter codes for H.264/AVC

Three-letter code	Syntactic structure in H.264/AVC
BSN	byte_stream_nal_unit
NALU	
NAL	nal_unit
STB	rbsp_slice_trailing_bits
RTB	rbsp_trailing_bits
RBSPs	
AUD	access_unit_delimiter_rbsp
ESQ	end_of_seq_rbsp
EST	end_of_stream_rbsp
FLD	filler_data_rbsp
PPS	pic_parameter_set_rbsp
SEI	sei_rbsp
SPS	seq_parameter_set_rbsp
SLA	slice_data_partition_a_layer_rbsp
SLB	slice_data_partition_b_layer_rbsp
SLC	slice_data_partition_c_layer_rbsp
SLW	slice_layer_without_partitioning_rbsp
Slice header	
DRP	dec_ref_pic_marking
PWT	pred_weight_table
RPL	ref_pic_list_reordering
SLH	slice_header
Slice data	
MBL	macroblock_layer
MBP	mb_pred
RES	residual

Three-letter code	Syntactic structure in H.264/AVC
CAB	residual_block_cabac
CAV	residual_block_cavlc
SLD	slice_data
SMP	sub_mb_pred
SEI	
BUP	buffering_period
DRR	dec_ref_pic_marking_repetition
FLP	filler_payload
FFF	full_frame_freeze
FFR	full_frame_freeze_release
FRS	full_frame_snapshot
MCS	motion_constrained_slice_group_set
PSR	pan_scan_rect
PIT	pic_timing
PRE	progressive_refinement_segment_end
PRS	progressive_refinement_segment_start
REP	recovery_point
RSM	reserved_sei_message
SCI	scene_info
SEM	sei_message
SEP	sei_payload
SPP	spare_pic
SSC	sub_seq_characteristics
SSI	sub_seq_info
SSL	sub_seq_layer_characteristics
UDR	user_data_registered_itu_t_t2
UDU	user_data_unregistered
VUI	
HRD	hrd_parameters
VUI	vui_parameters

If appropriate, these are paired together in the form: [TTT:LLL] where the following are used to generate the data given in the Trace file:

- TTT indicates the top called syntactic structure
- MMM indicates the middle called syntactic structure
- LLL indicates the lowest called syntactic structure

Three-letter codes for MPEG-4 and H.263

Three-letter code	Name in H.263	Name in MPEG-4	Standard section
[SC]	Picture Start Code PSC	Start Code	6.2, table 6.3
[VOS]	-	VisualObjectSequence	6.2.2
[VO]	-	VisualObject	6.2.2
[VOL]	-	VideoObjectLayer	6.2.3
[PL]	Picture Layer	-	
[GOV]	-	Group_of_VideoObjectPlane	6.2.4
[VOP]	-	VideoObjectPlane	6.2.5
[VPS]	Picture Layer	video_plane_with_short_header	6.2.5.2
[GOB]	GOB (Group of Blocks) layer	gob_layer	6.2.5.2
[MB]	MB (MacroBlock) layer	macroblock	6.2.6.2
[MOV]	MVD, MVD$_{2-4}$	motion_vector	6.2.6.2
[BLK]	Block layer	block	6.2.7
[inf]	Information about the internal state of the decoder or variables that may be described in the relevant		

Three-letter codes for MPEG-2

Three-letter code	Syntactic structure in H.264/AVC
Top level	
EUD	extension_and_user_data
GPH	group_of_pictures_header
PCD	picture_data
PCH	picture_header
PCX	picture_coding_extension
SQH	sequence_header
SQX	sequence_extension
VSQ	video_sequence

Three-letter code	Syntactic structure in H.264/AVC
Components	
APS	additional_pan_scan_parameters
ARW	active_region_window
BLK	block
CBP	coded_block_pattern
CDD	content_description_data
CPL	coded_picture_length
CTC	capture_timecode
EXD	extension_data
FCT	frame_or_field_capture_timestamp
MBK	macroblock
MBM	macroblock_modes
MVS	motion_vectors
NSC	next_start_code
PAD	padding_bytes
SLI	slice
USD	user_data
VEC	motion_vector
Extensions	
CRX	copyright_extension
CPX	camera_parameters_extension
ITU	itu_t_extension
PDX	picture_display_extension
PSS	picture_spatial_scalable_extension
PTS	picture_temporal_scalable_extension
QMX	quant_matrix_extension
SDX	sequence_display_extension
SSX	sequence_scalable_extension

If appropriate, these are paired together in the form: [TTT:LLL] where the following are used to generate the data given in the Trace file:

- TTT indicates the top called syntactic structure and

- MMM indicates the middle called syntactic structure(s) and
- LLL indicates the lowest called syntactic structure

Explanation of Bit/Bit Start

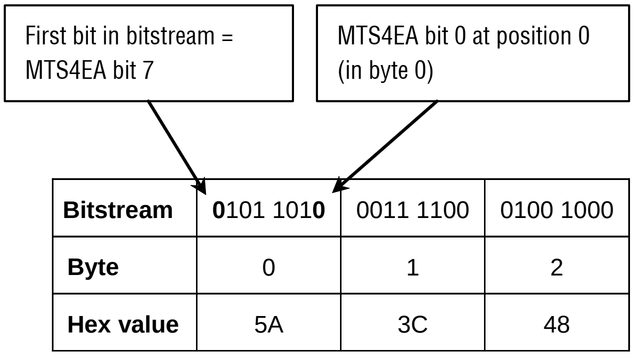
In each of the compression standards there are many bitstream fields which give values which are least significant bit first (lsbf) and many others which are most significant bit first (msbf). There are also many others that are simply bit-patterns: they do not have a direct numeric value.

MTS4EA has adopted the convention of naming the left-most bit as bit 7; as this is correct for presentation of numbers in bytes where the left-most bit is the most significant bit of the byte.

For msbf bitstream fields, numbering bit 7 as the first bit in each byte is correct.

For lsbf bitstream fields, the bit order has to be reversed when calculating the value given (and of course, all other decoders have to do this, not just MTS4EA).

To explain this further, consider a hypothetical bitstream:


First bit in bitstream = MTS4EA bit 7
MTS4EA bit 0 at position 0 (in byte 0)

Bitstream	**0**101 101**0**	0011 1100	0100 1000
Byte	0	1	2
Hex value	5A	3C	48

Therefore a bitstream field, of which the following statements are true, covers the above bitstream:

- The position is 1
- The bit/bit start is 4
- The length is 7 bits

Bitstream	0101 1010	001**1 1100**	**0**100 1000
Byte	0	1	2
Hex value	5A	3C	48

This field has a hexadecimal value of 0x71 if msbf or 0x47 if lsbf.

Bitstream syntax debugging using the MTS4EA

Using the various tools in the MTS4EA, bitstream syntax debugging can be quick and straightforward.

How the MTS4EA does Syntax Checking

The MTS4EA checks for hundreds of potential errors to ensure that the bitstream under test complies with the relevant standard. The MTS4EA checks for errors at multiple levels:

- At the first level, it will check for the presence of a particular allowable bitstream field (for example, if the picture size should be specified, that data consistent with this is present within the bitstream).
- At the second level, it will check within the bitstream field that the value given is permitted, that is, the bitstream field is self-consistent. (For example, that the values given for picture size are within the permitted range of values).
- At the third level, it will check that the values across multiple bitstream fields are consistent (for example, that there are the correct number of MacroBlocks for the picture size).

Principal Tools within the MTS4EA for Syntax Debugging

There are three main tools for syntax checking:

- The Alerts provided when displaying the video sequence using the MTS4EA (see description of alert levels in the *How to Use the MTS4EA* section).
- The Trace/Parse Bitstream function (see *How to Use the MTS4EA* section).
- The Trace/Interpret function (see *How to Use the MTS4EA* section).

Essentially, of the three principal tools provided within the MTS4EA (Alerts, Trace/Parse Bitstream and Trace/Interpret):

- Alerts give a summary of all levels of debug (first, second and third, as mentioned in *How the MTS4EA does Syntax* Checking)
- Trace/Parse Bitstream concentrates on the first and second levels
- Trace/Interpret concentrates on the second and third levels

All three tools are provided so that:

- A quick overview is available (using the Alerts)
- Sometimes it is useful to be able to continue debugging at a lower level, even if some higher-level errors are found (which is the reason for Trace/Parse Bitstream as well as Trace/Interpret)

NOTE. *You can open multiple windows in the MTS4EA, so that one window has the video, with windows for the Trace/Parse Bitstream output and a third window for Trace/Interpret output, all on the same video sequence.*

Procedure for bitstream syntax debugging

The following is a suggested procedure for getting started with debugging the video bitstreams you have generated.

After a period of time, you will generate your own methodologies, but this is a good starting point.

General Tips

Start debugging with short video sequences. It is easier to debug a shorter video sequence than a longer one, so start debugging with shorter versions of your compressed video sequences.

If you have to Trace out all the data of a whole video sequence into a Trace file, this file will not be too (unmanageably) large.

Use the HexView. It is often useful to also view the video bitstream file in the HexView provided within the MTS4EA, looking at the same point as the displayed video or a printed trace.

To use the HexView, refer to the *How to Use the MTS4EA* section.

Overview

In general, it is better to follow the following order:

1. Alerts
2. Analysis/Trace/Parse Bitstream (with or without the HexView)
3. Analysis/Trace/Interpret

However, for some debugging it is more useful to use Trace/Interpret first, followed by Parse Bitstream, if required, depending on the nature of the errors.

For each of these steps, use the appropriate procedure:

Alerts Debugging

1. Open the video sequence under test (using File/Open stream...)
2. The Alerts level is set to All when any video stream is opened.
3. If the syntax of the header of the bitstream is (reasonably) correct, then there will not be any Error or Warning messages; in which case, proceed to step 5.

4. If there are Error or Warning messages at this point, read the message carefully, noting the position (byte address) and bit position given. (See *How to Use the MTS4EA* for information on how to interpret the data provided in these pop-up windows.) At this point, it is likely you will need to understand the bits in the bitstream - go to *Trace/Parse Bitstream Debugging* below.

5. If there are no Errors or Warnings on File/Open, play the video sequence and see if any Errors or Warnings occur. If they do not, go to *Trace/Parse Bitstream Debugging* below.

6. If there are Errors and/or Warnings, stop playing the video, turn on the Summary Tooltip (see *How to Use the MTS4EA* section) and play the video again until it stops at the first Error/Warning.

7. Note the Frame number at which the Error/Warning occurs, and the other data (or open multiple copies of MTS4EA), then go to *Trace/Parse Bitstream Debugging* below.

Trace/Parse Bitstream Debugging

Procedure.

1. If you know the frame number at which an Error or Warning occurs, set the Frame Range to one frame before and one frame after the location of the Error or Warning. (See *Trace Enable, Frame range tab* in the *How to Use the MTS4EA* section).

2. Set the Trace filename and the maximum Trace File size (see *How to Use the MTS4EA* section).

3. Set Parse Bitstream on and all other Trace functions off.

4. Turn Summary Tooltip on (icon) and play the video from the start; you can stop the video as soon as it has passed the frame numbers to be captured.

5. Use View trace... to see the Trace file output.

6. You should see a Trace out file similar to that shown in the see *How to Use the MTS4EA* section.

7. Search for Errors and Warnings (using the Find next button in the View trace... window).

8. If there are no Errors or Warnings, it is a higher-level error. Go to *Trace/Interpret Debugging* on page 7–13.

9. If there are Errors/Warnings, examine the bitstream data and the mnemonic given on the right side of the trace file and relate this to the video standard you are using to see if you can discover the problem.

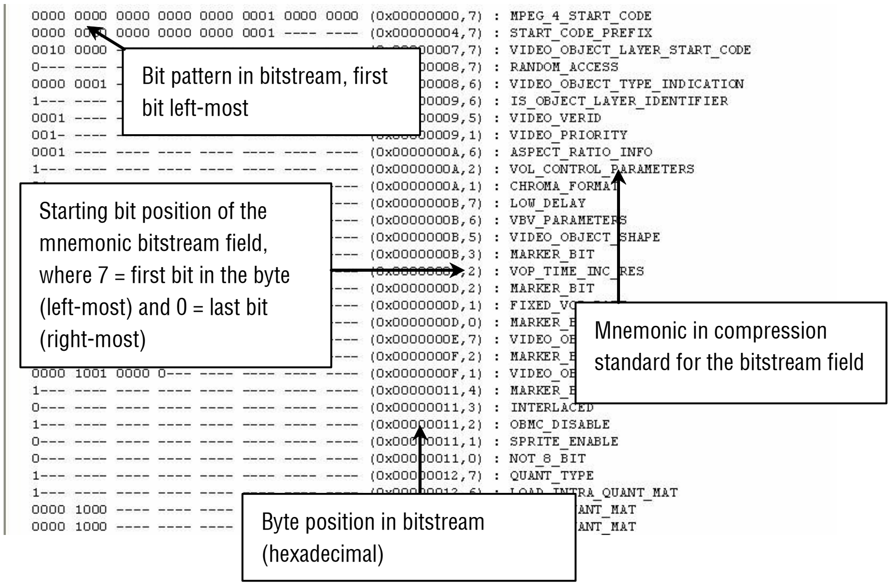
***NOTE.** It is very important to understand that the error may well have occurred at some prior point in the bitstream. It could be that the error/warning reported is the ripple effect result of some earlier incorrect value. This is usually the case with an Out of Sync error.*

For example:

- In MPEG-4, if using video_plane_with_short_header() (MPEG-4 standard section 6.2.5.2)
- And pei = 1, then following this should be eight bits of a psupp field (these can be repeated), followed by the gob_layer(), gob_resync_marker (17 bits, 0000 0000 0000 0000 1)
- However, if pei = 0, then the gob_resync_marker should follow directly
- Therefore if pei was set to 0 by accident, but there was a psupp field inserted set to all zeroes, the bit pattern following pei would be:
  0000 0000 0000 0000 0000 0000 1
- This would appear to be an error in the gob_resync_marker (that is because the 17th bit after pei=0 would not be set to 1)

10. Check that there are no elements of a standard being used that should not be there. For example, if you are compressing data to H.263 baseline or MPEG-4 short_header, search for: `{263+}` and `{263 Ann}` to find any H.263+ annexes used.
11. Check the three-letter codes in the square brackets used by MTS4EA and reported in the trace output, such as [SC], as given in *Three-letter codes for H.264/AVC* on page 7–2, and that the three-letter code corresponds with the relevant standard section.

Parse Bitstream example outputs. See also Parse Bitstream in the *How to Use the MTS4EA* section for more information.



```
0000 0000 0000 0000 0000 0001 0000 0000 (0x00000000,7) : MPEG_4_START_CODE
0000 0000 0000 0000 0000 0001 ---- ---- (0x00000004,7) : START_CODE_PREFIX
0010 0000 -                                   00007,7) : VIDEO_OBJECT_LAYER_START_CODE
0--- ---- -                                   00008,7) : RANDOM_ACCESS
0000 0001 -                                   00008,6) : VIDEO_OBJECT_TYPE_INDICATION
1--- ---- -                                   00009,6) : IS_OBJECT_LAYER_IDENTIFIER
0001 ---- -                                   00009,5) : VIDEO_VERID
001- ---- -                                   00009,1) : VIDEO_PRIORITY
0001 ---- ---- ---- ---- ---- ---- ---- (0x0000000A,6) : ASPECT_RATIO_INFO
1--- ---- ---- ---- ---- ---- ---- ---- (0x0000000A,2) : VOL_CONTROL_PARAMETERS
                                    --- (0x0000000A,1) : CHROMA_FORMAT
                                    --- (0x0000000B,7) : LOW_DELAY
                                    --- (0x0000000B,6) : VBV_PARAMETERS
                                    --- (0x0000000B,5) : VIDEO_OBJECT_SHAPE
                                    --- (0x0000000B,3) : MARKER_BIT
                                    --- (0x0000000C,2) : VOP_TIME_INC_RES
                                    --- (0x0000000D,2) : MARKER_BIT
                                    --- (0x0000000D,1) : FIXED_VO
                                    --- (0x0000000D,0) : MARKER_B
                                    --- (0x0000000E,7) : VIDEO_OB
                                    --- (0x0000000F,2) : MARKER_B
0000 1001 0000 0--- ---- ---- ---- ---- (0x0000000F,1) : VIDEO_OB
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000011,4) : MARKER_B
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000011,3) : INTERLACED
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000011,2) : OBMC_DISABLE
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000011,1) : SPRITE_ENABLE
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000011,0) : NOT_8_BIT
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000012,7) : QUANT_TYPE
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000012,6) : LOAD_INTRA_QUANT_MAT
0000 1000 ---- ---- ----                                         ANT_MAT
0000 1000 ---- ---- ----                                         ANT_MAT
```

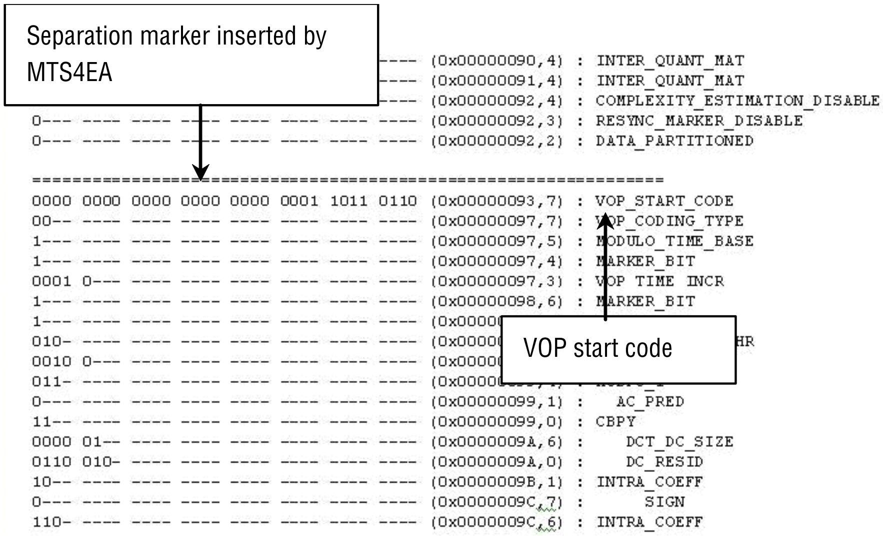
Bit pattern in bitstream, first bit left-most

Starting bit position of the mnemonic bitstream field, where 7 = first bit in the byte (left-most) and 0 = last bit (right-most)

Mnemonic in compression standard for the bitstream field

Byte position in bitstream (hexadecimal)

MPEG-4 example, at start of bitstream

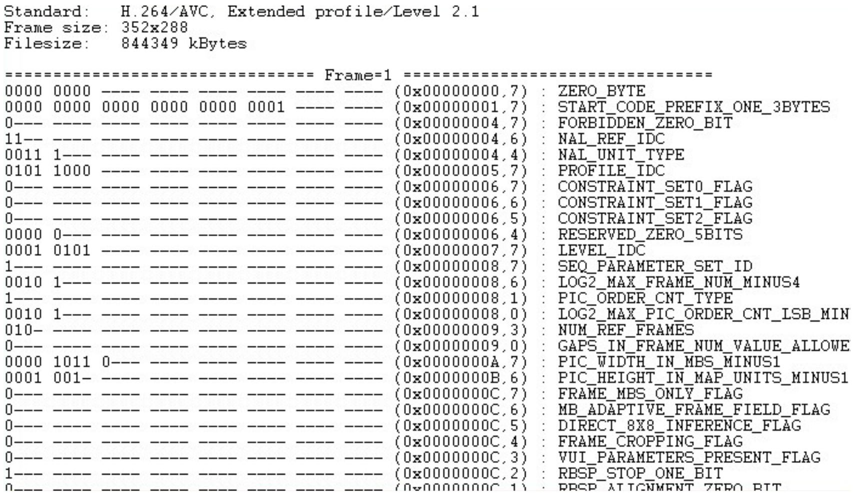


```
                                   ---- (0x00000090,4) : INTER_QUANT_MAT
                                   ---- (0x00000091,4) : INTER_QUANT_MAT
                                   ---- (0x00000092,4) : COMPLEXITY_ESTIMATION_DISABLE
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000092,3) : RESYNC_MARKER_DISABLE
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000092,2) : DATA_PARTITIONED

==================================================================
0000 0000 0000 0000 0000 0001 1011 0110 (0x00000093,7) : VOP_START_CODE
00-- ---- ---- ---- ---- ---- ---- ---- (0x00000097,7) : VOP_CODING_TYPE
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000097,5) : MODULO_TIME_BASE
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000097,4) : MARKER_BIT
0001 0--- ---- ---- ---- ---- ---- ---- (0x00000097,3) : VOP TIME INCR
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000098,6) : MARKER_BIT
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000
010- ---- ---- ---- ---- ---- ---- ---- (0x00000                       HR
0010 0--- ---- ---- ---- ---- ---- ---- (0x00000
011- ---- ---- ---- ---- ---- ---- ---- (0x00000
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000099,1) :   AC_PRED
11-- ---- ---- ---- ---- ---- ---- ---- (0x00000099,0) : CBPY
0000 01-- ---- ---- ---- ---- ---- ---- (0x0000009A,6) :    DCT_DC_SIZE
0110 010- ---- ---- ---- ---- ---- ---- (0x0000009A,0) :    DC_RESID
10-- ---- ---- ---- ---- ---- ---- ---- (0x0000009B,1) : INTRA_COEFF
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000009C,7) :      SIGN
110- ---- ---- ---- ---- ---- ---- ---- (0x0000009C,6) : INTRA_COEFF
```

Separation marker inserted by MTS4EA

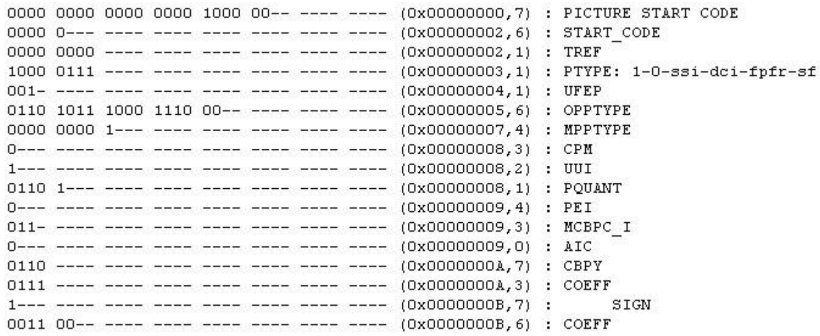
VOP start code

MPEG-4 example, at start of VOP



```
Standard:   H.264/AVC, Extended profile/Level 2.1
Frame size: 352x288
Filesize:   844349 kBytes

================================ Frame=1 ================================
0000 0000 ---- ---- ---- ---- ---- ---- (0x00000000,7) : ZERO_BYTE
0000 0000 0000 0000 0000 0001 ---- ---- (0x00000001,7) : START_CODE_PREFIX_ONE_3BYTES
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000004,7) : FORBIDDEN_ZERO_BIT
11-- ---- ---- ---- ---- ---- ---- ---- (0x00000004,6) : NAL_REF_IDC
0011 1--- ---- ---- ---- ---- ---- ---- (0x00000004,4) : NAL_UNIT_TYPE
0101 1000 ---- ---- ---- ---- ---- ---- (0x00000005,7) : PROFILE_IDC
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000006,7) : CONSTRAINT_SET0_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000006,6) : CONSTRAINT_SET1_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000006,5) : CONSTRAINT_SET2_FLAG
0000 0--- ---- ---- ---- ---- ---- ---- (0x00000006,4) : RESERVED_ZERO_5BITS
0001 0101 ---- ---- ---- ---- ---- ---- (0x00000007,7) : LEVEL_IDC
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000008,7) : SEQ_PARAMETER_SET_ID
0010 1--- ---- ---- ---- ---- ---- ---- (0x00000008,6) : LOG2_MAX_FRAME_NUM_MINUS4
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000008,1) : PIC_ORDER_CNT_TYPE
0010 1--- ---- ---- ---- ---- ---- ---- (0x00000008,0) : LOG2_MAX_PIC_ORDER_CNT_LSB_MIN
010- ---- ---- ---- ---- ---- ---- ---- (0x00000009,3) : NUM_REF_FRAMES
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000009,0) : GAPS_IN_FRAME_NUM_VALUE_ALLOWE
0000 1011 0--- ---- ---- ---- ---- ---- (0x0000000A,7) : PIC_WIDTH_IN_MBS_MINUS1
0001 001- ---- ---- ---- ---- ---- ---- (0x0000000B,6) : PIC_HEIGHT_IN_MAP_UNITS_MINUS1
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,7) : FRAME_MBS_ONLY_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,6) : MB_ADAPTIVE_FRAME_FIELD_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,5) : DIRECT_8X8_INFERENCE_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,4) : FRAME_CROPPING_FLAG
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,3) : VUI_PARAMETERS_PRESENT_FLAG
1--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,2) : RBSP_STOP_ONE_BIT
0--- ---- ---- ---- ---- ---- ---- ---- (0x0000000C,1) : RBSP_ALIGNMENT_ZERO_BIT
```

H.264/AVC example, at start of byte stream



```
0000 0000 0000 0000 1000 00-- ---- ---- (0x00000000,7) : PICTURE START CODE
0000 0--- ---- ---- ---- ---- ---- ---- (0x00000002,6) : START_CODE
0000 0000 ---- ---- ---- ---- ---- ---- (0x00000002,1) : TREF
1000 0111 ---- ---- ---- ---- ---- ---- (0x00000003,1) : PTYPE: 1-0-ssi-dci-fpfr-sf
001- ---- ---- ---- ---- ---- ---- ---- (0x00000004,1) : UFEP
0110 1011 1000 1110 00-- ---- ---- ---- (0x00000005,6) : OPPTYPE
0000 0000 1--- ---- ---- ---- ---- ---- (0x00000007,4) : MPPTYPE
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000008,3) : CPM
1--- ---- ---- ---- ---- ---- ---- ---- (0x00000008,2) : UUI
0110 1--- ---- ---- ---- ---- ---- ---- (0x00000008,1) : PQUANT
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000009,4) : PEI
011- ---- ---- ---- ---- ---- ---- ---- (0x00000009,3) : MCBPC_I
0--- ---- ---- ---- ---- ---- ---- ---- (0x00000009,0) : AIC
0110 ---- ---- ---- ---- ---- ---- ---- (0x0000000A,7) : CBPY
0111 ---- ---- ---- ---- ---- ---- ---- (0x0000000A,3) : COEFF
1--- ---- ---- ---- ---- ---- ---- ---- (0x0000000B,7) :      SIGN
0011 00-- ---- ---- ---- ---- ---- ---- (0x0000000B,6) : COEFF
```

H.263 example, at start of bitstream

Trace/Interpret Debugging

Overview. This is the highest level of debugging; MTS4EA interprets the bitstream and tries to tie it up with allowable values in the standards.

There are many occasions on which Trace/Parse Bitstream will not give an error/warning but Trace Interpret will. For example:

- Parse Bitstream will find a bitstream field and check that it is in the permissible range of values

- But Interpret will ensure that the value given is consistent with the rest of the data in the bitstream

There are times when it is useful to use both.

Procedure. The procedure for Trace/Interpret debugging is very similar to that for Trace/Parse Bitstream:

1. If you know the frame number at which an Error or Warning occurs, then set the Frame Range to one frame before and one frame after the location of the Error or Warning. (see *Trace Enable, Frame range tab* in the *How to Use the MTS4EA* section)
2. .Set the Trace File name and the maximum Trace File size (see *How to Use the MTS4EA* section).
3. Set Interpret on and all other Trace functions off.
4. Turn the Summary Tooltip on and play the video from the start; you can stop the video as soon as it has passed the frame numbers to be captured.
5. Use View trace... to see the Trace file output.
6. You should see a Trace out file similar to that shown in the see How to Use the MTS4EA section.
7. Search for Errors and Warnings (using the Find next button in the View trace... window).
8. If there are Errors/Warnings/Out of Syncs, then examine the bitstream data and the mnemonic given on the right side of the trace file and relate this to the video standard you are using to see if you can discover the problem.

***NOTE.** It is very important to understand that the error may have occurred at some prior point in the bitstream. It could be that the error/warning reported is the ripple effect result of some earlier incorrect value. This is usually be the case with an Out of Sync error.*

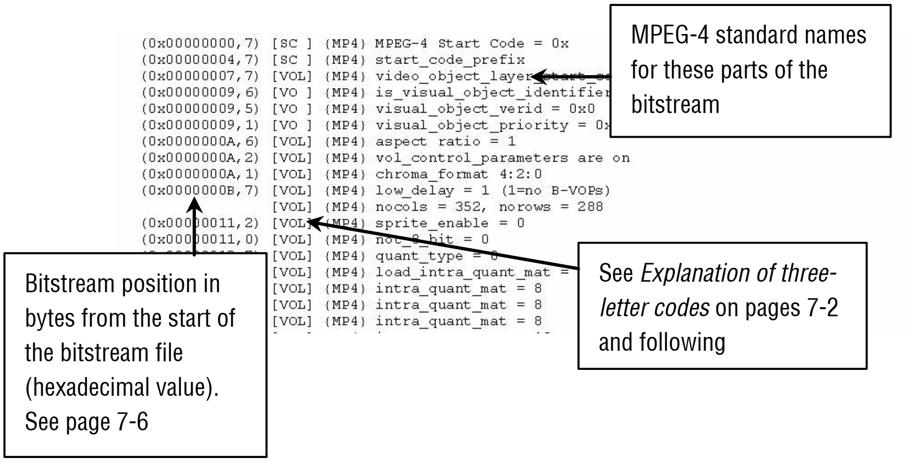
For example:

- In MPEG-4, if using video_plane_with_short_header() (MPEG-4 standard section 6.2.5.2)
- And pei = 1 then following this should be 8 bits of a psupp field (these can be repeated), followed by the gob_layer(), gob_resync_marker (17 bits, 0000 0000 0000 0000 1)
- However, if pei = 0 then the gob_resync_marker should follow directly
- Therefore if pei was set to 0 by accident, but there was a psupp field inserted set to all zeroes, the bit pattern following pei would be:
  0000 0000 0000 0000 0000 0000 1
- This would appear to be an error in the gob_resync_marker (that is because the 17th bit after pei=0 would not be set to 1)

9. Check that there are no elements of a standard being used that should not be there. For example, if you are compressing data to H.263 baseline or MPEG-4 short_header, search for: `{263+}` and `{263 Ann}` to find any H.263+ annexes used.

10. Check the three-letter codes in the square brackets used by the MTS4EA and reported in the trace output, such as [SC], as given in *Three-letter codes for H.264/AVC* on page 7–2, and that the three-letter code corresponds with the relevant standard section.

Interpret example outputs. See also *Interpret* in the *How to Use the MTS4EA* section for more information.



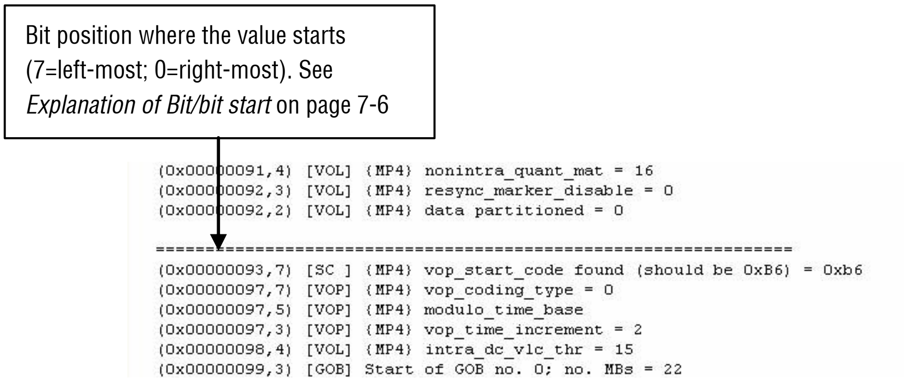
```
(0x00000000,7) [SC ] {MP4} MPEG-4 Start Code = 0x
(0x00000004,7) [SC ] {MP4} start_code_prefix
(0x00000007,7) [VOL] {MP4} video_object_layer
(0x00000009,6) [VO ] {MP4} is_visual_object_identifier
(0x00000009,5) [VO ] {MP4} visual_object_verid = 0x0
(0x00000009,1) [VO ] {MP4} visual_object_priority = 0x
(0x0000000A,6) [VOL] {MP4} aspect ratio = 1
(0x0000000A,2) [VOL] {MP4} vol_control_parameters are on
(0x0000000A,1) [VOL] {MP4} chroma_format 4:2:0
(0x0000000B,7) [VOL] {MP4} low_delay = 1 (1=no B-VOPs)
               [VOL] {MP4} nocols = 352, norows = 288
(0x00000011,2) [VOL] {MP4} sprite_enable = 0
(0x00000011,0) [VOL] {MP4} not_8_bit = 0
               [VOL] {MP4} quant_type = 0
               [VOL] {MP4} load_intra_quant_mat =
               [VOL] {MP4} intra_quant_mat = 8
               [VOL] {MP4} intra_quant_mat = 8
               [VOL] {MP4} intra_quant_mat = 8
```

MPEG-4 standard names for these parts of the bitstream

See *Explanation of three-letter codes* on pages 7-2 and following

Bitstream position in bytes from the start of the bitstream file (hexadecimal value). See page 7-6

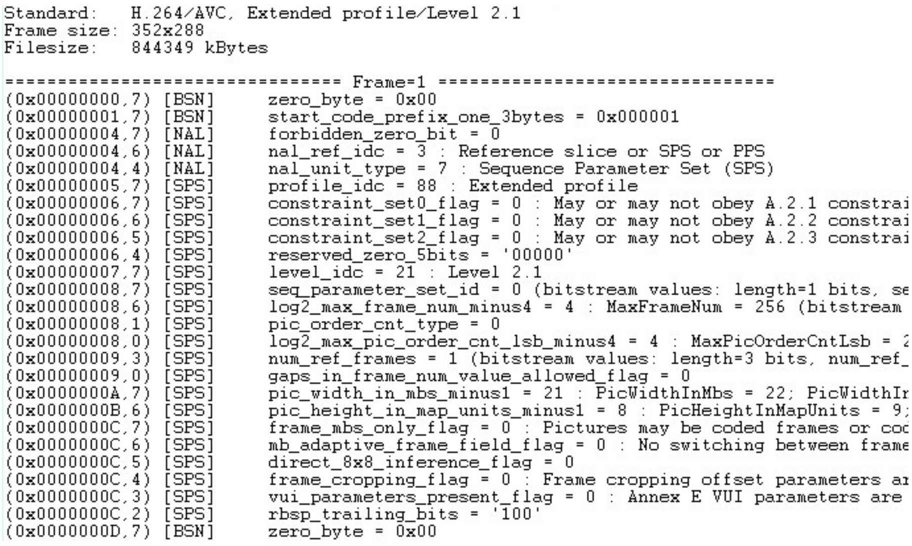
MPEG-4 Example, at start of bitstream



Bit position where the value starts (7=left-most; 0=right-most). See *Explanation of Bit/bit start* on page 7-6

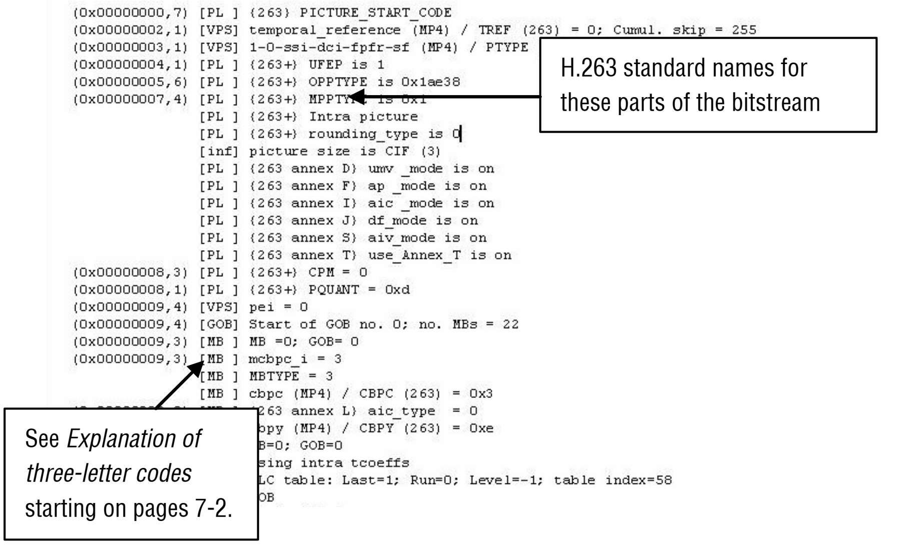
```
(0x00000091,4) [VOL] {MP4} nonintra_quant_mat = 16
(0x00000092,3) [VOL] {MP4} resync_marker_disable = 0
(0x00000092,2) [VOL] {MP4} data partitioned = 0

===============================================================
(0x00000093,7) [SC ] {MP4} vop_start_code found (should be 0xB6) = 0xb6
(0x00000097,7) [VOP] {MP4} vop_coding_type = 0
(0x00000097,5) [VOP] {MP4} modulo_time_base
(0x00000097,3) [VOP] {MP4} vop_time_increment = 2
(0x00000098,4) [VOL] {MP4} intra_dc_vlc_thr = 15
(0x00000099,3) [GOB] Start of GOB no. 0; no. MBs = 22
```

MPEG-4 example, at start of VOP



```
Standard:   H.264/AVC, Extended profile/Level 2.1
Frame size: 352x288
Filesize:   844349 kBytes

=============================== Frame=1 ===============================
(0x00000000,7) [BSN]     zero_byte = 0x00
(0x00000001,7) [BSN]     start_code_prefix_one_3bytes = 0x000001
(0x00000004,7) [NAL]     forbidden_zero_bit = 0
(0x00000004,6) [NAL]     nal_ref_idc = 3 : Reference slice or SPS or PPS
(0x00000004,4) [NAL]     nal_unit_type = 7 : Sequence Parameter Set (SPS)
(0x00000005,7) [SPS]     profile_idc = 88 : Extended profile
(0x00000006,7) [SPS]     constraint_set0_flag = 0 : May or may not obey A.2.1 constrai
(0x00000006,6) [SPS]     constraint_set1_flag = 0 : May or may not obey A.2.2 constrai
(0x00000006,5) [SPS]     constraint_set2_flag = 0 : May or may not obey A.2.3 constrai
(0x00000006,4) [SPS]     reserved_zero_5bits = '00000'
(0x00000007,7) [SPS]     level_idc = 21 : Level 2.1
(0x00000008,7) [SPS]     seq_parameter_set_id = 0 (bitstream values: length=1 bits, se
(0x00000008,6) [SPS]     log2_max_frame_num_minus4 = 4 : MaxFrameNum = 256 (bitstream
(0x00000008,1) [SPS]     pic_order_cnt_type = 0
(0x00000008,0) [SPS]     log2_max_pic_order_cnt_lsb_minus4 = 4 : MaxPicOrderCntLsb = 2
(0x00000009,3) [SPS]     num_ref_frames = 1 (bitstream values: length=3 bits, num_ref_
(0x00000009,0) [SPS]     gaps_in_frame_num_value_allowed_flag = 0
(0x0000000A,7) [SPS]     pic_width_in_mbs_minus1 = 21 : PicWidthInMbs = 22; PicWidthIr
(0x0000000B,6) [SPS]     pic_height_in_map_units_minus1 = 8 : PicHeightInMapUnits = 9;
(0x0000000C,7) [SPS]     frame_mbs_only_flag = 0 : Pictures may be coded frames or coc
(0x0000000C,6) [SPS]     mb_adaptive_frame_field_flag = 0 : No switching between frame
(0x0000000C,5) [SPS]     direct_8x8_inference_flag = 0
(0x0000000C,4) [SPS]     frame_cropping_flag = 0 : Frame cropping offset parameters ar
(0x0000000C,3) [SPS]     vui_parameters_present_flag = 0 : Annex E VUI parameters are
(0x0000000C,2) [SPS]     rbsp_trailing_bits = '100'
(0x0000000D,7) [BSN]     zero_byte = 0x00
```

H.264/AVC example, at start of byte stream



```
(0x00000000,7) [PL ] {263} PICTURE_START_CODE
(0x00000002,1) [VPS] temporal_reference (MP4) / TREF (263) = 0; Cumul. skip = 255
(0x00000003,1) [VPS] 1-0-ssi-dci-fpfr-sf (MP4) / PTYPE
(0x00000004,1) [PL ] {263+} UFEP is 1
(0x00000005,6) [PL ] {263+} OPPTYPE is 0x1ae38
(0x00000007,4) [PL ] {263+} MPPTYPE is 0x1
               [PL ] {263+} Intra picture
               [PL ] {263+} rounding_type is 0
               [inf] picture size is CIF (3)
               [PL ] {263 annex D} umv _mode is on
               [PL ] {263 annex F} ap _mode is on
               [PL ] {263 annex I} aic _mode is on
               [PL ] {263 annex J} df_mode is on
               [PL ] {263 annex S} aiv_mode is on
               [PL ] {263 annex T} use_Annex_T is on
(0x00000008,3) [PL ] {263+} CPM = 0
(0x00000008,1) [PL ] {263+} PQUANT = 0xd
(0x00000009,4) [VPS] pei = 0
(0x00000009,4) [GOB] Start of GOB no. 0; no. MBs = 22
(0x00000009,3) [MB ] MB =0; GOB= 0
(0x00000009,3) [MB ] mcbpc_i = 3
               [MB ] MBTYPE = 3
               [MB ] cbpc (MP4) / CBPC (263) = 0x3
                      263 annex L} aic_type  = 0
                      bpy (MP4) / CBPY (263) = 0xe
                      B=0; GOB=0
                      sing intra tcoeffs
                      LC table: Last=1; Run=0; Level=-1; table index=58
                      OB
```

H.263 standard names for these parts of the bitstream

See *Explanation of three-letter codes* starting on pages 7-2.

H.263 example at start of bitstream

Appendices

Appendix A: Decoder Plugins for MTS4EA

This section covers use of MTS4EA decoder Plugins.

Purpose of MTS4EA Decoder Plugins

The MTS4EA Decoder Plugins allow the advanced user to substitute various elements of MTS4EA decoder with custom/alternate elements.

There are a number of reasons why you may want to do this:

- There are various divergences between the outputs of different decoders due to rounding errors or different floating-point arithmetic used, because the video standards do not completely define these. The differences given mean that the output given by the MTS4EA - particularly using the YUV output in batch mode (see *Command Line/Batch Mode* in the *How to Use the MTS4EA* section for more information) - cannot be exactly compared with the output from another decoder. By being able to substitute a user's own part of the codec for that section of the decoder, the YUV outputs generated should exactly match those of the user. See *`idct` Plugin* on page A–4 for an example.

- Sometimes the video standard has errors and differs from common implementations. Many implementations differ from the standard in this way. See *`qs` Plugin* on page A–4 for an example.

- Sometimes it is useful to be able to use the MTS4EA function within your own decoder, to guarantee that numerically identical results are output by the MTS4EA decoder and your own.

***NOTE.** The DLLs provided by **Tektronix** are provided and licensed solely for the purposes of test and may **NOT** be redistributed under **ANY** circumstances.*

Support of Decoder Plugins

Use of the Decoder Plugins assumes a considerable and detailed understanding of the video standard(s) concerned, as well as a thorough understanding of how to write Windows DLLs, and are provided for use by advanced developers only.

***NOTE.** Other than the documentation provided here and in the files provided on the CD in the folder of the Decoder Plugins, no technical support will be provided to a developer in developing their own Decoder Plugin.*

Decoder Plugin File Layout on the CD

All the Decoder Plugins are on the CD in the folder: `\Decoder plugins`

Below this are three folders:

`\bin`	: contains the DLLs
`\docs`	: contains the documentation
`\include`	: contains the .h files to include in your programs

bin folder

Below this folder are the DLLs provided by the MTS4EA. The DLLs are organized in their own folders.

Where a DLL is not dependent upon a particular video standard, this folder resides directly below the `bin` folder. Where there are standards-dependencies, then the DLL folders reside below the folder of the standard concerned.

Examples are:

Folder	Files	Description

`\Decoder plugins\bin\idct\`	`vpIdct.dll`
	`vpIdct.exp`
	`vpIdct.lib`
		IDCT DLL file
		} export LIB files to link to
		} the IDCT DLL

`\Decoder plugins\bin\mpeg4\gmc`	`vpGmc.dll`
	`vpGmc.exp`
	`vpGmc.lib`
		GMC (Global Motion Compensation) DLL file
		} export LIB files to link to
		} the GMC DLL

`\Decoder plugins\bin\mpeg4\qs`	`vpQs.dll`
	`vpQs.exp`
	`vpQs.lib`
		Quarter Sample (QS) DLL file
		} export LIB files to link to
		} the QS DLL

docs folder

This contains the documentation for each supplied DLL.

To view the documentation, open the file: `index.html` using a browser.

The API to the DLLs, file lists, structures, data fields and globals are all accessed by clicking on the related links.

include folder

This folder contains all the `.h` files that contain the declarations of the interface for each plugin.

For each plugin, only one include is required: that is for the related DLL For example, for the IDCT plugin, only the `vpIdct.h` file needs to be included in your own source files.

NOTE. *Any other `.h` files that are listed in the include folder, and which are also needed by the included `.h` file, will be included within the `.h` file.*

An example of this is the ‘vpIdct.h’, which includes ‘MTS4EA.h’.

Format for Generating Decoder Plugins

The Decoder Plugins were written using Microsoft Visual C 6.0.

The Plugins are all Microsoft Windows DLLs.

The only versions of Windows supported are given in the *Software, Hardware, and User Prerequisites* section.

NOTE. *No other development tools or variants to these are supported for the Decoder Plugins.*

Use of Decoder Plugins

To use a specific Plugin, copy it from the plugin folder to the folder where the MTS4EA executable is (typically `C:\Program files\Tektronix\MTS4EA\`).

There will already be a DLL of the same name within the MTS4EA executable folder; the copied Plugin should overwrite this.

As delivered, all the Plugins that are in the MTS4EA executable folder are MTS4EA variants of these.

Decoder Plugins Provided

`idct` Plugin

This allows you to substitute your own IDCT function instead of using the one provided with the MTS4EA.

By using the same IDCT function in the MTS4EA and in your own decoder, the YUV outputs from the MTS4EA should be numerically identical to the YUV outputs provided by your own decoder.

qs Plugin

The MPEG-4 standard (Reference [1], see *Compression Standards and File Types* section) specifies the recommended method for calculating quarter sample accuracy predicted blocks (relating to motion vectors). However, the standard is nonoptimal in the way that these are calculated (for example introducing more rounding errors than necessary) and is not as clear as it might be. Consequently, common implementations of the quarter sample functions implement the calculations slightly differently (typically in a different order) to that given in the standard.

The Microsoft reference decoder (Reference [7], see *Compression Standards and File Types* section), although it is (theoretically) normative, is one such implementation that does not actually implement the quarter sample calculations in accordance with the standard; it takes a better approach.

There has been some debate about this within the MPEG committee, and the prevailing conclusion seems to be that the Microsoft implementation may be the preferred version and that the standard should at some point be changed to reflect this.

The MTS4EA uses the version as specified in the standard, and the MTS4EA will continue to track the standard and so will change only if the standard is changed. However, you can substitute your own quarter sample Plugin DLL, to use your own or emulate the one used by Microsoft.

gmc Plugin

This allows you to substitute your own GMC (Global Motion Compensation) function instead of using the one provided with the MTS4EA.

The Microsoft reference decoder software (Reference [7], see *Compression Standards and File Types* section) differs from the MPEG-4 standard (Reference [1], see *Compression Standards and File Types* section) in several respects, for example, rounding differences and differences in coordinate calculations. This means that the results of the YUV outputs from a Microsoft decoder will be different than the standard if GMC is used. (These differences are reflected in the Microsoft encoder, so that the Microsoft software is self-consistent.) There is no indication from the MPEG-4 committee of any planned change in the standard. This is simply a variance between the standard and the reference software.

MTS4EA follows the standard. However if you want to follow the Microsoft version of the implementation of GMC or use your own, then you may substitute your own GMC DLL.

Appendix B: Tests of MTS4EA

This section covers a report on the MTS4EA tests of the MPEG-4 Normative bitstreams and Donated bitstreams (see *Compression Standards and File Types* section).

Tests with MPEG-4 normative and donated bitstreams

Many of the MPEG-4 Normative bitstreams and Donated bitstreams have errors: some of these errors are clear non-conformance to the standard; others are errors in the data encoded in the bitstreams.

Also, in a number of cases, the Microsoft reference decoder software will not decode these bitstreams at all or does so incorrectly.

These divergences are listed below. Many of them are recognized and documented by the MPEG committee as errors, but they have yet to be corrected in the Standard and/or bitstreams and/or Microsoft reference decoder software.

Notes on versions used:

- MTS4EA: v2.0.0.0
- Microsoft reference software: FDAM1-2.3-001213 version 2 dated July 3rd 2000 (note: this reference software has been updated since these tests, so the current reference software may behave differently)
- MPEG-4 Normative ISO bitstreams: dated 05/11/2001, see *Compression Standards and File Types* section
- MPEG-4 Donated bitstreams: referred to in section 4.5.8 in document N3067 dated 1999-03-18; streams dated 14/07/2000, see *Compression Standards and File Types* section

n/a = not applicable

Bitstreams: Normative ISO

Stream name	Decodes with ?	Notes
vcon-ge1	n/a	Interlace not allowed in Simple Profile
vcon-ge2	n/a	Interlace not allowed in Simple Profile
vcon-ge3	n/a	Interlace not allowed in Simple Profile
vcon-ge4	n/a	Interlace not allowed in Simple Profile
vcon-ge6	n/a	Interlace not allowed in Simple Profile
vcon-ge8	n/a	OBMC not allowed in Simple, Advanced Simple, or Main Profiles
vcon-ge10	n/a	Interlace not allowed in Simple Profile
vcon-ge11	n/a	Interlace not allowed in Simple Profile
vcon-ge12	n/a	Interlace not allowed in Simple Profile
vcon-ge13-L1	Yes	
vcon-ge13-L2	Yes	
vcon-ge13-L3	Yes	
vcon-ge14	n/a	OBMC not allowed in Simple, Advanced Simple, or Main Profiles
vcon-ge16-L1	Yes	MS reference software cannot decode
vcon-ge16-L2	Yes	MS reference software cannot decode
vcon-ge16-L3	Yes	MS reference software cannot decode
vcon-ge18	n/a	Interlace not allowed in Simple Profile
vcon-ge19	n/a	OBMC not allowed in Simple, Advanced Simple, or Main Profiles
vcon-ge23	n/a	Interlace not allowed in Simple Profile
vcon-ge24	n/a	OBMC not allowed in Simple, Advanced Simple, or Main Profiles
vcon-ge25	n/a	OBMC not allowed in Simple, Advanced Simple, or Main Profiles

Bitstreams: Donated \ I-VOP

Stream name	Decodes with ?	Notes
hit000.m4v	Yes	Single frame. Time listed incorrectly in MPEG-4 part 4: time is actually 33 ms (vop_time_increment = 30).
jvc000.m4v	Yes	
mit000.m4v	Yes	Uses error resilience tool (data partitioning)
mit001.m4v	Yes	Uses error resilience tool (resynchronization)
mit002.m4v	Yes	Uses error resilience tool (resynchronization)
mit003.m4v	Yes	Uses error resilience tool (resynchronization)
mit004.m4v	Yes	Uses error resilience tool (data partitioning)
mit005.m4v	Yes	Uses error resilience tool (data partitioning)
mit006.m4v	Yes	Uses error resilience tool (data partitioning)
san000.m4v	Yes	
san001.m4v	Yes	

Bitstreams: Donated \ Overall

Stream name	Decodes with ?	Notes
hit016.m4v	Yes	
hit017.m4v	Yes	Visible artifacts are in bitstream (plays the same with MS software)
hit018.m4v	Yes	
hit019.m4v	Yes	
hit020.m4v	Yes	
hit021.m4v	Yes	
hit022.m4v	Yes	
hit023.m4v	Yes	
hit024.m4v	Yes	
mit030.m4v	Yes	
mit031.m4v	Yes	Uses error resilience tool (data partitioning)

Bitstreams: Donated \ Short Header

Stream name	Decodes with ?	Notes
hit031.m4v	Yes	
hit032.m4v	Yes	Visible errors but these are encoded in the bitstream (MS software plays the same)
hit033.m4v	Yes	
hit034.m4v	Yes	
hit035.m4v	Yes	MPEG committee reports that MS software fails to decode (not tested)
hit036.m4v	Yes	Not short_header compliant as Pspare is sent (H.263+ compliant)
hit037.m4v	Yes	
hit038.m4v	Yes	
hit039.m4v	Yes	
hit040.m4v	Yes	
jvc022.m4v	Yes	
jvc023.m4v	Yes	
jvc024.m4v	Yes	
jvc025.m4v	Yes	
mit020.m4v	Yes	
mit021.m4v	Yes	MPEG committee reports that MS software fails to decode (not tested)
mit022.m4v	Yes	MPEG committee reports that MS software fails to decode (not tested)
mit023.m4v	Yes	MPEG committee reports that MS software fails to decode (not tested)
mit024.m4v	Yes	MPEG committee reports that MS software fails to decode (not tested)
san021.m4v	Yes	
san022.m4v	Yes	Poor frames are encoded in bitstream (MS software plays the same)
san023.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)
san024.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)

Bitstreams: Donated \ P-VOP

Stream name	Decodes with ?	Notes
hit001.m4v	Yes	Some visible errors, for example, frame 1, GOB 2, MB 15 block Y3 but these are in bitstream (MS software plays the same)
hit002.m4v	Yes	Some visible errors, for example, in frame 2, GOB4, MB 1 and in frame 3 - several MBs down left side, for example, MB 0 at GOB 14, 15, 16 - these are errors in the encoded bitstream (MS software plays the same)
hit003.m4v	Yes	Errors in bitstream, for example, in MB 0 GOB 6 frame 4. It seems that MVs were not correctly calculated in encoded stream. (MS software plays the same)
hit004.m4v	Yes	
hit005.m4v	Yes	Some visible artifacts in bitstream (MS software plays the same)
hit006.m4v	Yes	
hit007.m4v	Yes	
hit008.m4v	Yes	
hit009.m4v	Yes	
hit010.m4v	Yes	
hit011.m4v	Yes	
hit012.m4v	Yes	
hit013.m4v	Yes	
hit014.m4v	Yes	
jvc001.m4v	Yes	
jvc002.m4v	Yes	
jvc003.m4v	Yes	
jvc004.m4v	Yes	
jvc005.m4v	Yes	
jvc006.m4v	Yes	
jvc007.m4v	Yes	
jvc008.m4v	Yes	
jvc009.m4v	Yes	
jvc010.m4v	Yes	
jvc011.m4v	Yes	
jvc012.m4v	Yes	
jvc013.m4v	Yes	

Stream name	Decodes with ?	Notes
jvc014.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)
jvc015.m4v	Yes	
jvc016.m4v	Yes	
jvc017.m4v	Yes	
jvc018.m4v	Yes	
jvc019.m4v	Yes	
jvc020.m4v	Yes	
jvc021.m4v	Yes	
mit007.m4v	Yes	Syntax error in bitstream: video_object_type_indicator is 0 at byte: 8. bit start=0. This is defined as a RESERVED value (MPEG-4 std. p.110 Table 6-10). Plays this stream, assuming it is Simple Profile (an Error message is given to this effect)
mit008.m4v	Yes	
mit009.m4v	Yes	Visible artifacts are encoded in bitstream (MS software plays the same)
mit010.m4v	Yes	Uses error resilience tool (data partitioning)
mit011.m4v	Yes	Visible artifacts are in bitstream, for example, frame 4 halo above head in GOB 0 MBs 3,4,5 and poor MBs with hand movement, for example, frame 4, GOB,MB: 10,8 11,8 (MS software plays the same)
mit012.m4v	Yes	Uses error resilience tool (data partitioning)
mit013.m4v	Yes	Uses error resilience tool (data partitioning). Stream not decoded correctly by MS software - occasional green lines in some MBs
mit014.m4v	Yes	Uses error resilience tool (data partitioning)
mit015.m4v	Yes	Uses error resilience tool (data partitioning)
mit016.m4v	Yes	Uses error resilience tool (data partitioning)
mit017.m4v	Yes	Uses error resilience tool (data partitioning)
mit018.m4v	Yes	Uses error resilience tool (data partitioning)
mit019.m4v	Yes	
san002.m4v	Yes	
san003.m4v	Yes	
san004.m4v	Yes	
san005.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)

Stream name	Decodes with ?	Notes
san006.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)
san007.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)
san008.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)
san009.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)
san010.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)
san011.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)
san012.m4v	Yes	Non-smooth motion is encoded in bitstream (MS software plays the same)
san013.m4v	Yes	
san014.m4v	Yes	
san015.m4v	Yes	Graininess, visible artifacts, and intermediate lower resolution frames are encoded in the bitstream (MS software plays the same)
san016.m4v	Yes	Visible artifacts and intermediate lower resolution frames are encoded in the bitstream (MS software plays the same)
san017.m4v	Yes	
san018.m4v	Yes	
san019.m4v	Yes	
san020.m4v	Yes	

Bitstreams: Donated \ Error

Stream name	Decodes with ?	Notes
hit025.m4v	Yes	Use of resync markers
hit026.m4v	Yes	Use of resync markers
hit027.m4v	Yes	Uses error resilience tool (data partitioning)
hit028.m4v	Yes	Uses error resilience tool (data partitioning)
hit029.m4v	Yes	Uses error resilience tool (data partitioning)
hit030.m4v	Yes	Uses error resilience tool (data partitioning)
mit025.m4v	Yes	Use of resync markers
mit026.m4v	Yes	Use of resync markers
mit027.m4v	Yes	Uses error resilience tool (data partitioning) Poor quality picture (MS software plays the same)
mit028.m4v	Yes	Uses error resilience tool (data partitioning)
mit029.m4v	Yes	Uses error resilience tool (data partitioning). Artifacts in frames 5 and 6 are errors in the encoded bitstream: seems to be motion vectors incorrectly calculated when the stream was encoded (MS software plays the same)

Glossary

Glossary

AC coefficient	Any DCT coefficient for which the frequency in one or both dimensions is non-zero
Annex	The optional methods of coding allowed in H263+
B-VOP	A VOP that is coded using motion compensated prediction from past and/or future reference VOPs
Backward motion vector	A motion vector that is used for motion compensation from a reference VOP at a later time in display order
Backward prediction	Prediction from the future reference VOP
Bitstream	A compressed data entity where each binary digit has a specific meaning that is defined by the compression standard
Block	An 8-row by 8-column matrix of samples, or 64 DCT coefficients (source, quantized or dequantized)
Buffer analysis	Analysis of use of processor, memory and/or other resources by a particular video decoder standard
Coded Block Pattern (cbp)	A word used in compression to identify which blocks of a MacroBlock are coded
Coefficients	The Discrete Cosine Transform transforms data from a spatial domain (pixels or pixel differences) into the frequency domain, because it makes the data easier to compress. In the standards used here, transforms turn a block of 8x8 pixel data (or pixel differences) into a block of 8x8 transform coefficients
CSV file	A file format which contains data separated by commas (Comma Separated Variable)
DC coefficient	The DCT coefficient for which the frequency is zero in both directions

DCT	Discrete Cosine Transform - the mathematical transform that all these compression standards use as their basis
DCT coefficient	Amplitude of the specific DCT basis function
Dequantization	Process of rescaling the quantized DCT coefficients after their representation in the bitstream has been decoded and before they are presented to the inverse DCT
Filter	A mathematical transform designed to remove certain frequencies from a signal. Here mainly used either within the coding loop (as in H.261) to try and avoid some of the coding artifacts and reduce bit-rate, or used as a post-process to improve the subjective quality
Frame	An individual picture from a video sequence
Forward motion vector	A motion vector that is used for motion compensation from a reference frame VOP at an earlier time in the display order
Forward prediction	Prediction from a past reference VOP
Global Motion Compensation	Use of global spatial transformation to improve the efficiency of the prediction of sample values by providing offsets into the past reference VOPs containing previously decoded sample values that are used to form the prediction error
GMC	Global Motion Compensation
GOB	Group Of Blocks - an entity defined within some of the standards in order to subdivide a frame into more manageable units
Histogram	A graph of the frequency of occurrence of a variable
I-VOP, intra-coded VOP	A VOP coded using information only from itself
Intra coding	Coding of a MacroBlock or VOP using information only from that MacroBlock or VOP
MacroBlock, MB	Basic coding unit of the standards used in this program. It consists of four blocks of 8x8 luminance data (arranged in a 16x16 manner) together with the two chrominance components

U and V, which are also 8x8 blocks, but which cover the same area of the picture as the 16x16 luminance pixels - part of the compression is that chrominance can be sampled at a lower frequency than luminance

MacroBlock Type

The mode, according to the Standard, in which the MacroBlock is encoded

Mode

Classification of the coding type of the MacroBlock

Motion Vector

Two-dimensional vector that points from the current MacroBlock to an area in the previous frame that is used to predict the current data

Motion compensation

Use of motion vectors to improve the efficiency of the prediction of sample values, where the motion vectors provide offsets into the past and/or future reference VOPs containing previously decoded sample values that are used to form the prediction error

Motion estimation

Process of estimating motion vectors during the encoding process

Parse Bitstream

The process of parsing a bitstream into the constituent words that are allowed within the standard

Quantizer

The discrete value that is used to reduce the amount of information present in the DCT of a block. It can vary from 1 to 31 in most standards, where 1 is the finest level (most accurate coding) and 31 is the coarsest level (least accurate coding)

Quantization matrix

Set of sixty-four 8-bit values used by the dequantizer

Quantized DCT coefficients

DCT coefficients before Dequantization, represented in variable-length coded form in the bitstream

Quantizer scale

Scale factor coded form in the bitstream and used by the decoding process to scale the dequantization

Slice

A subdivision of a picture that is used as a unit of encoding, as used in H.263 and MPEG-2

SEI	Supplemental enhancement information
VOP	Effectively a frame of video (MPEG-4)

Index

Index