User Manual

Tektronix

MTS4EA
Compressed Video ES Analyzer
071-1641-01

This document supports software version 4.0.

www.tektronix.com

Contacting Tektronix

Tektronix, Inc.
14200 SW Karl Braun Drive or 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.

Table of Contents

Introduction

Applications

Software, Hardware and User Prerequisites

Installation and Licensing

Compression Standards and File Types

How to Use MTS4EA

Bitstream Syntax Debugging

Appendix A: Decoder Plugins for MTS4EA

Appendix B: Tests of MTS4EA

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 MTS4EA

  ***NOTE**. Even if you read nothing else from this manual, read the TUTORIAL section. It provides the basics on how to use and how to get the benefit from the software.*

- 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/

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\`

At various points in the manual you will see Notes, or items that you should be aware of when running MTS4EA.

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

Online Help

Online help is provided, to give easy access to the information within this user manual. This can be accessed by any of the following:

- Pressing the F1 key
- Selecting Help topics (F1) from the menu
- Clicking the Help icon  on the toolbar

User Manual and Tutorials

The information in this manual is also provided in 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 sometimes it is 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 data that has been encoded using the H.264/AVC, MPEG-4, MPEG-2, H.263+, H.263, and H.261 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 AD951A or AD953A MPEG Test Systems.

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

- Next Generation (H.264/AVC, MPEG-4 and 3GPP) and Legacy (MPEG-2, H.261, 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
- Real-time and non real-time decoding and analysis of compressed video elementary stream files for display on UI (dependant on PC performance)
- Buffer Analysis
- Reference to Encoded video visual difference display
- PSNR Measurements
- Batch mode to allow automated testing
- YUV decoded video output to file for baseband video analysis

Applications

Applications

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

Equipment Manufacturers

- Semiconductor device designers and manufacturers
- Video 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 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

Applications

- Analyze the performance (qualitative and quantitative) of compression algorithms, to assist with optimization, comparison, and evaluation
- Check conformity and correctness of a compressed bit-stream against the various standards
- Confirm interoperability between different vendors compression algorithms
- Provide comprehensive statistics that will enable the informed user to optimize 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 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 2-4 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; where ~200 MB RAM is reserved for the RAM disk. (This would allow a 200 MB Trace file to be output to the RAM disk.)

AD951A and AD953A MPEG Test Systems

You can install MTS4EA on either a Tektronix AD951A or AD953A (software version 6.7 or later). The installation procedure is described in the next section.

MTS300 MPEG Test and Monitoring

You can install MTS4EA on a Tektronix MTS300. The installation procedure is described in the next section.

User Prerequisites

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

It therefore assumes that the user of the software understands these video standards in significant detail and can therefore interpret the information presented by MTS4EA.

A Glossary is provided, and a list of references (**Standards** References on page 5–20).

MTS4EA Standalone Software Options

Item	Opt	Description	
MTS4EA		Next Generation Elementary Stream Analyzer. Must order either Option 4EAB or 4EAF, and the required dongle type.	
MTS4EA	4EAB	Base software with video standard package including: MPEG-4 Simple Profile, H.263+, H.263, H.261, TS Extraction, Single local user license. Includes English manual and CD	Mandatory and Mutually Exclusive
MTS4EA	4EAF	Base software with floating license token, one needed for each floating License required. Need to order the same quantity of each option as floating licenses. Includes English manual and CD	
MTS4EA	PPD	Parallel Port Dongle	Mandatory and Mutually Exclusive
MTS4EA	USB	USB Dongle	
Other standalone software options			
MTS4EA	M4SP	MPEG-4 Advanced Simple Profile (Levels 0 -5)	
MTS4EA	M2ML	MPEG-2 Main Profile Main Level	
MTS4EA	M2HL	MPEG-2 Main Profile High Level & High Level 1440 (High Definition)	
MTS4EA	AVCE	H.264/AVC Baseline and Extended Profiles (Levels 1 – 5)	
MTS4EA	AVCM	H.264/AVC Main Profile (Levels 1 - 5)	
MTS4EA	AVCH	H.264/AVC High Profile with FRExt (10 bit, 4:2:2:, 4:4:4)	
MTS4EA	VC1	VC-1 (all Profiles, all Levels) and ASF extraction	
MTS4EA	AUD	Audio playout (incl. MPEG2 Layer 1 & 2, AAC, HE AAC)	
MTS4EA	SWS	First 12 Months Software Subscription when MTS4EA is ordered as a standalone package, if floating then the number of SWS options must equal the number of 4EAF options ordered.	
MTS4EAUP		Upgrades for MTS4EA Software including upgrade instructions, and dongle upgrade disk. See options for more details.	
MTS4EAUP	DDP	Dongle Upgrade Disk - Parallel, CD and Manual,	Mandatory

Item	Opt	Description	
MTS4EAUP	DDU	Dongle Upgrade Disk - USB, CD and Manual,	and Mutually Exclusive
MTS4EAUP	4EAB	Base software with video standard package including: MPEG-4 Simple Profile, H.263+, H.263, H.261, TS extraction. (MTS4x0 & AD953A dongle upgrade only). Single local user license	
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, H.261, TS extraction.	
MTS4EAUP	FLUP	Upgrade a current single local user license to floating. If more than one floating license needed then must order MTS4EAUP 4EAF to add extra licenses. Only available for V4.0and higher. If upgrading more than one License to a combined floating license then all dongles but one must be returned to Tektronix within 30 days of receipt of the dongle upgrade disk. By not returning these you agree to pay Tektronix for the additional licenses as new, and an invoice will be raised for these. Also all Current MTS4EAs must be V4.0 and have identical options.	
Other Standalone Software Upgrade options			
MTS4EAUP	M4SP	MPEG-4 Advanced Simple Profile (Levels 0 -5)	
MTS4EAUP	M2ML	MPEG-2 Main Profile Main Level	
MTS4EAUP	M2HL	MPEG-2 Main Profile High Level & High Level 1440 (High Definition)	
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)	
MTS4EAUP	VC1	VC-1 (all Profiles, all Levels) and ASF extraction	
MTS4EAUP	AUD	Audio playout (incl. MPEG2 Layer 1 & 2, AAC, HE AAC)	
MTS4EAUP	SWS	First 12 Months Software Subscription Service when installed on an existing Tektronix Analyser (cannot be ordered with MTS4EAUP Opt. UPG)	
MTS4EAUP	SWS1	Software Subscription in the 2nd or subsequent years.	
MTS4EAUP	UPG	Upgrade to latest version of MTS4EA base software and installed options. Includes CD and Manual (Requires MTS4EAUP Opt SWS1)	
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.	

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 dongle with the correct license permissions.

***NOTE.** The dongle is very important as it contains the **MTS4EA** license and therefore embodies the value of **MTS4EA**. If the dongle is lost, you will have to purchase another dongle, 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 hardware dongle is installed. When using node-locked licenses, each PC that is to run MTS4EA must have a separate node-locked license.

Dongle-based floating license

The dongle-based floating license uses a hardware dongle located on a license server to provide the license key.

Note the following system requirements:

- The dongle 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. Subsequently, the installation of the MTS4EA (with either a node-locked license or floating license) is similar.

The floating license system enables a single hardware key (the DES DK2 dongle) to serve up licenses to multiple users connected over a network. The dongle is pre-programmed with licenses prior to 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 hardware key, which contains one or more licenses. It is possible to have several license servers on the network each with their own hardware 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:

1. Select a server machine that will host the hardware key.

2. Connect the dongle to the server (either via the USB port or parallel port depending on dongle type).

3. On the server machine, insert the MTS4EA installation CD.

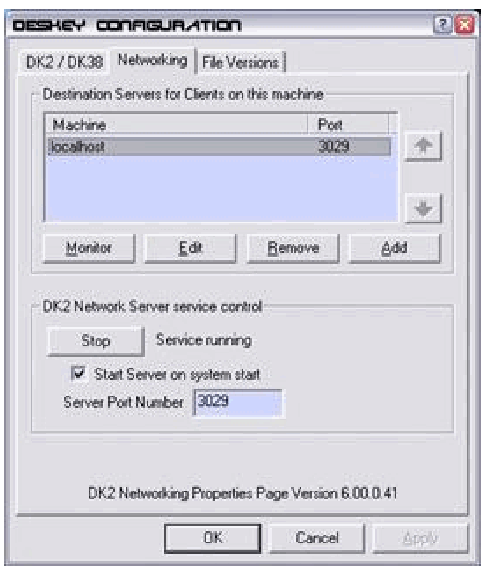
   ***NOTE**. When the CD is inserted, the MTS4EA setup screen will be displayed. Select the Close button at the bottom of the window to terminate the MTS4EA installation program.*

4. Using Windows Explorer, locate and open the `DES` directory. Run the program `dnsrv32.exe`. During the installation process accept the default entries.

5. Reboot the server machine if requested to do so at the end.

6. 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 below (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.*



7. 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, so 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. Note that this does not cause the server to reboot; it simply 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.)

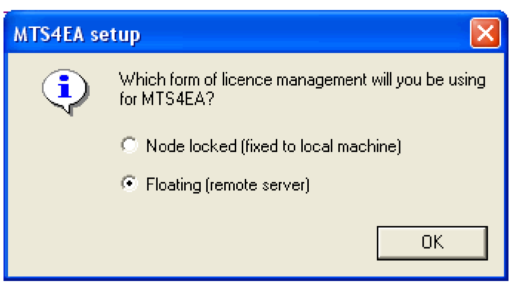
1. Insert the MTS4EA installation CD.

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

---

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

3. Select the license management option required.



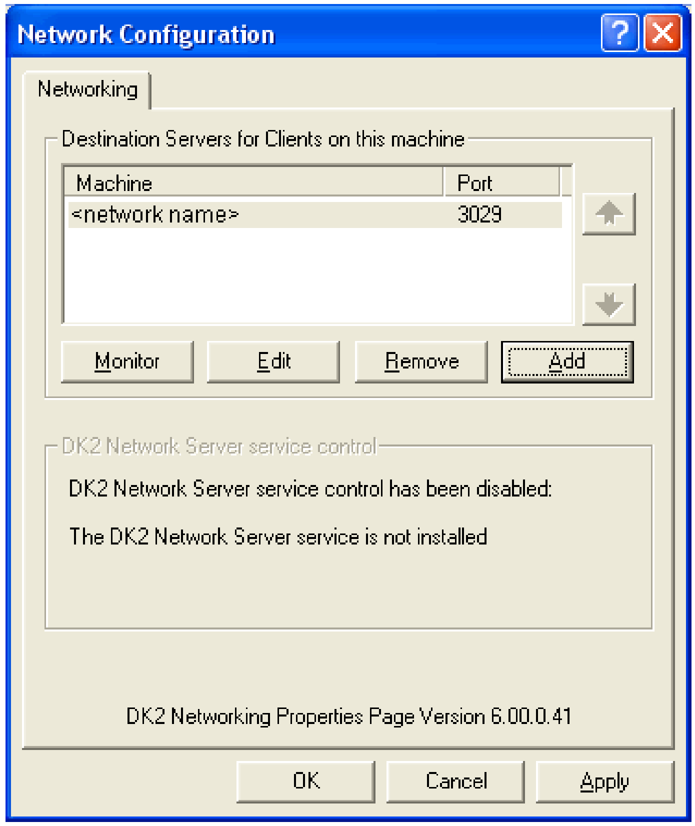
Select OK to continue.
If Node locked is chosen proceed to step 8.
If floating is chosen proceed to the next step.

4. An initial dialog is shown. Press Next. The DK2 dongle driver is installed.

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

---

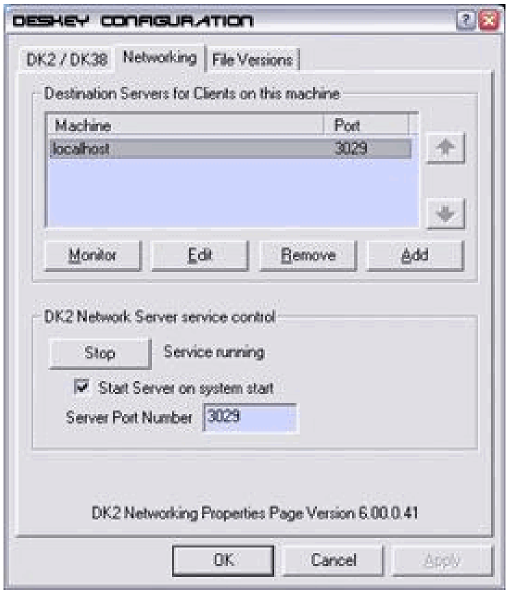
5. At the end of the driver installation setup, a Network Configuration dialog will appear (shown below). This enables the server(s) installed in the previous section to be added to the list of machines to query for licenses.



6. 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 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 in the License Server section above. The actual name of the machine can be used if preferred.



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

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

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

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

Testing the License Server

To ensure the previous installations have worked, perform the following steps:

1. Remove any Tektronix DK2 dongles from the client machine and ensure the server machine has a suitably programmed dongle connected.
2. 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 dongle is inserted correctly and the DK2 Network Server service is running. The latter can be checked by opening the DESkey Configuration dialog 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 dongle 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 dongle check 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 dongle, try connecting the dongle to the client machine and run the MT4EA software again (ensure the Enable Local DK2 Access checkbox is ticked in the DK2/DK38 tab of the DESkey configuration dialog). If this works, the dongle is working so try another server machine. If this does not work please 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 first diagram above).

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 upon which MTS4EA is based

Supported Video Compression Standards

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

- H.264/AVC
- VC-1
- MPEG-4 Part 2
- MPEG-2 Part 2
- H.263+, H.263
- H.261

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

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

MTS4EA will also open uncompressed video files: see *Opening an* uncompressed video file (any file extension) on page 6–25 for more information.

MTS4EA will also open audio files to the standards given in **Supported** Audio Compression Standards on page 5–11, and container files which have these types of audio within them.

NOTE. *Note that not all container files support all the types of audio listed under* ***Supported*** *Audio Compression Standards on page 5–11: 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 5–20).

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
  - High Profile, all Levels from 1 to 5.1 inclusive
  - High/10-bit, all Levels from 1 to 5.1 inclusive
  - High/4:2:2, all Levels from 1 to 5.1 inclusive
  - High/4:4:4, all Levels from 1 to 5.1 inclusive

H.264/AVC Tools

(See Reference [13] under **Standards** References on page 5–20 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	1	0 or 1	0 or 1	0 or 1
▪ 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 (L 1-2 & 4.2 - 5.1)	Yes (L 1-2 & 4.2 - 5.1)	Yes (L 1-2 & 4.2 - 5.1)	Yes (L 1-2 & 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

Tool	Baseline	Extended	Main	High	High 10	High 4:2:2	High 4:4:4
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	Yes	Yes	Yes	Yes	Yes	Yes	Yes
▪ CABAC	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
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 5–20).

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

(See Reference [17] under **Standards** References on page 5–20.)

MPEG-4

This version of MTS4EA supports the following elements of the MPEG-4 standard (see Reference [1] under **Standards** References on page 5–20).

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 when compared with 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 5–20)
  - Level 4
  - Level 5
- Main Profile:
  - Level 2

NOTE. *Level 0 is an addition to Simple Profile which is not in the MPEG-4 standard Reference [1] under* ***Standards*** *References on page 5–20. 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

(grey box)	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 5–20).

MPEG-2 Profiles, Levels

The following Profiles and Levels are supported, subject to the restrictions given under **Standards** References on page 5–20:

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

MPEG-2 tools

For MPEG-2 Main Profile @ Main Level, @ High Level and @ 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
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 5–20)
- H.263+ (see Reference [2] under **Standards** References on page 5–20), 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.

H.261

The following H.261 standard is supported:

- H.261 - baseline standard (see Reference [3] under **Standards** References on page 5–20).

Supported Audio Compression Standards

MTS4EA supports audio extraction and playout. Supported standards are:

- MPEG-1/2 part 3 audio (Reference [19] under **Standards** References on page 5–20)
- MPEG-2 part 7 AAC (Reference [20] under **Standards** References on page 5–20)
- MPEG-4 AAC (AAC plus) (Reference [21] under **Standards** References on page 5–20)
- MPEG-4 HE AAC (Reference [22] under **Standards** References on page 5–20)

NOTE. *Full details of these audio standards are not provided, as MTS4EA is primarily a video analysis tool.*

See **Permitted** Audio Types/Formats on page 5–19 for details of the container file types that may have audio in and are supported by MTS4EA.

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] under **Standards** References on page 5–20.

RCV files

RCV files contain video in the VC-1 format. See Reference [17] under **Standards** References on page 5–20.

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 MTS4EA analyzes should conform to the relevant part of the MPEG-4 standard – Reference [11] under **Standards** References on page 5–20.

By default, 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...* *Ctrl+O on page* 6–13.

The structure of the MP4 file can be examined, using the View file structure... option on the Analysis menu: see **View** file structure... Ctrl+R on page 6–158 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] under **Standards** References on page 5–20.

By default, 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...* Ctrl+O on page 6–13.

The structure of the 3GPP file can be examined, using the View file structure... option on the Analysis menu: see **View** file structure... Ctrl+R on page 6–158 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, 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] under **Standards** References on page 5–20.

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] under **Standards** References on page 5–20), 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 amongst a number of VOB files.

NOTE. The user can specify the different VOBs that contain a single Elementary Stream, so that 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] under **Standards** References on page 5–20.

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.

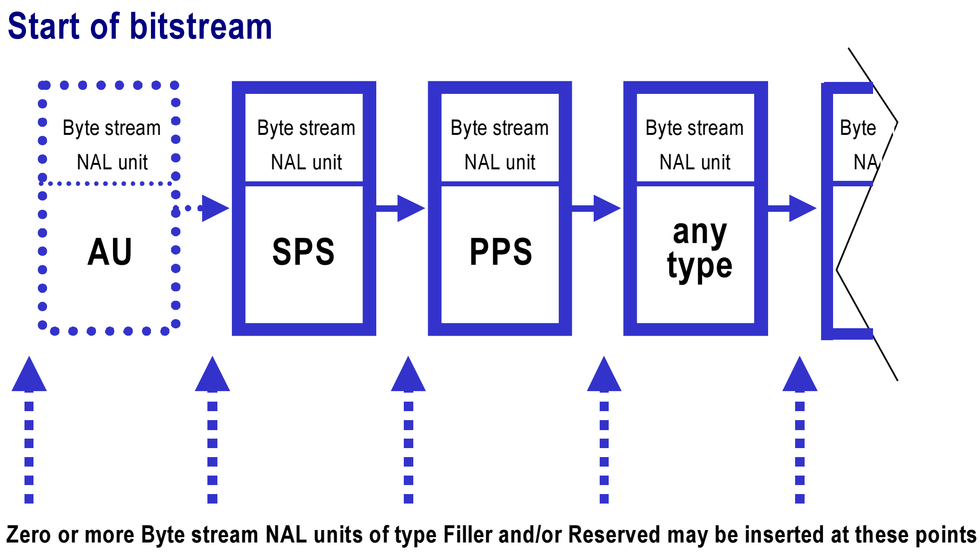
Once the video is extracted, MTS4EA will automatically determine the format of the compressed video in the file by searching through the video and looking for the appropriate start codes. Accordingly the file name does not have to have a particular file extension to be decoded correctly.

H.264/AVC Byte Stream file format

In accordance with the H.264/AVC standard (see Reference [13] under **Standards** References on page 5–20), the video file must start as given in the diagram overleaf.

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] under **Standards** References on page 5–20), 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] under **Standards** References on page 5–20 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] under **Standards** References on page 5–20), 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) under **Standards** References on page 5–20 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] under **Standards** References on page 5–20), the Elementary Stream file will start with a 32-bit start code: 0x000001b3 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.

H.261 file format

The first bits in an H.261 compressed file should be the 20-bit PSC (Picture Start Code): 0000 0000 0000 0001 0000.

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 the user computer (the limit to the size is actually 1 million Terabytes [2^{60}] 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/Formats

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

- MP4 (MPEG-4 Part 1 and Part 14)
- MPEG-2 Transport Stream
- MPEG-2 Program Stream

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 v1.13

[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

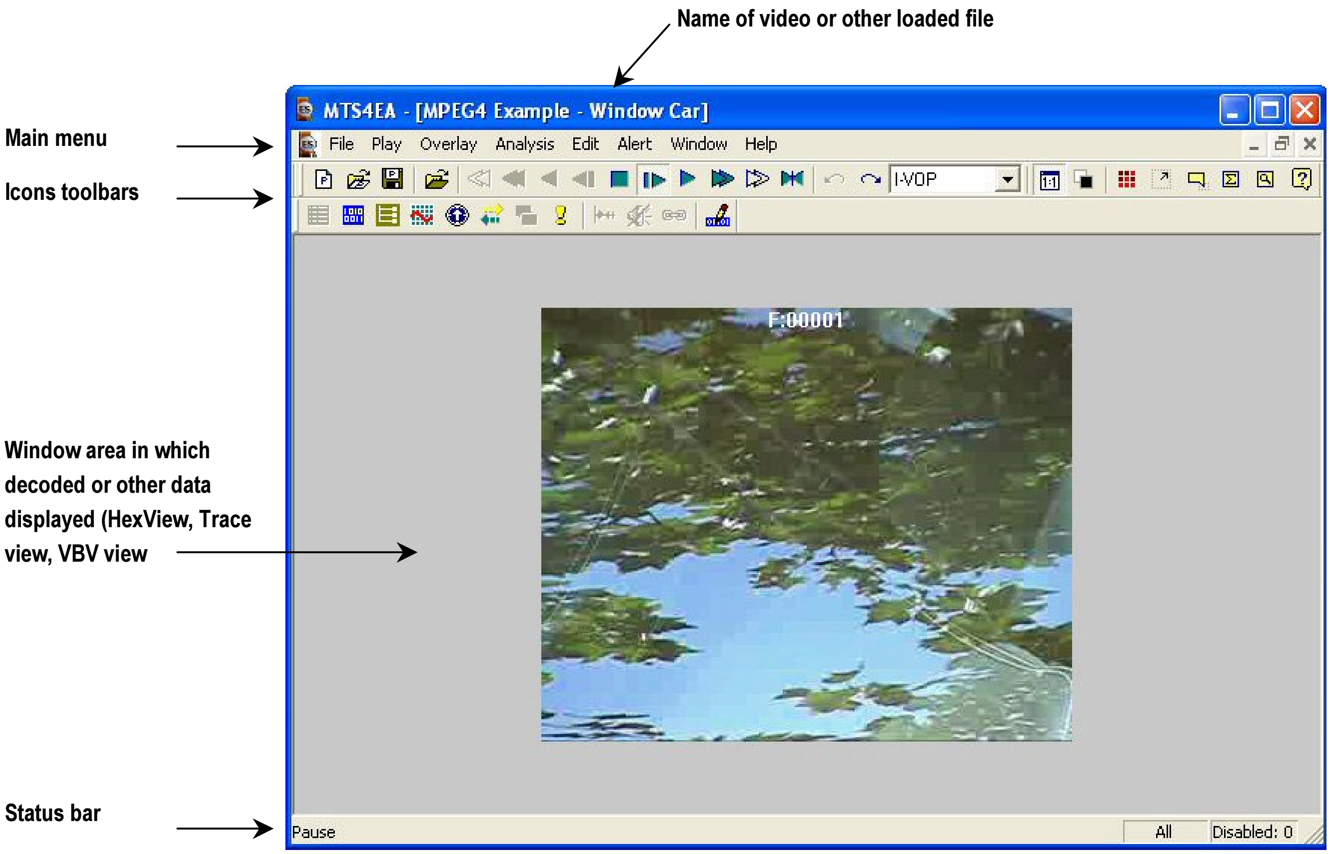
How to Use MTS4EA

How to Use 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 MTS4EA on page 6–3)
- Describes each of the menu items in detail and how to use them (see *Main Menu* on page 6–12 to **Help** Menu on page 6–206)
- Explains the icon toolbars, below the main menu (see *Icon Toolbars* on page 6–209)
- 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 6–213)
- Explains the information provided on the status bar (see **Status** Bar on page 6–216)
- Lists the shortcut keys, such as `Ctrl+A` to Pause/Frame Advance one frame (see **Ctrl** Shortcut Keys on page 6–218)
- Explains the use of the Alt key to select menu items (see **Alt** Menu Keys on page 6–219)
- Explains the use of the command line/batch mode (see *Command Line/Batch Mode* on page 6–220)

Window Elements

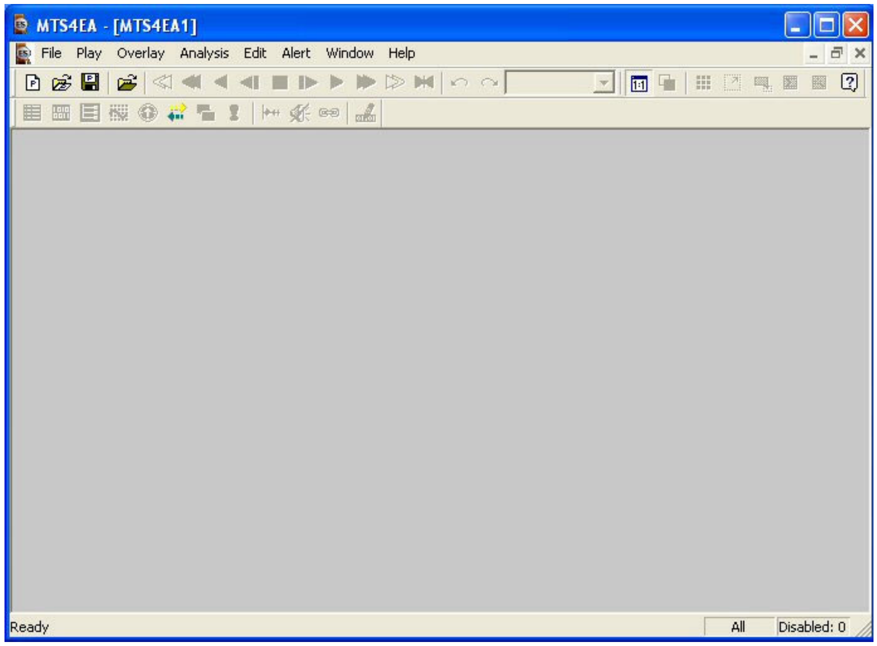


__NOTE__. The above screenshot was taken with MTS4EA running under Windows XP. When MTS4EA is running under Windows 2000 there may be some visual differences in the borders, menus, and the way that pop-up menus appear from those shown in the screenshots. However, there are no functional differences in the way 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 which are context-sensitive, depend upon the video standard being used and/or the explicit function being performed.

Starting to Use MTS4EA

When MTS4EA starts, the display initially looks as below:

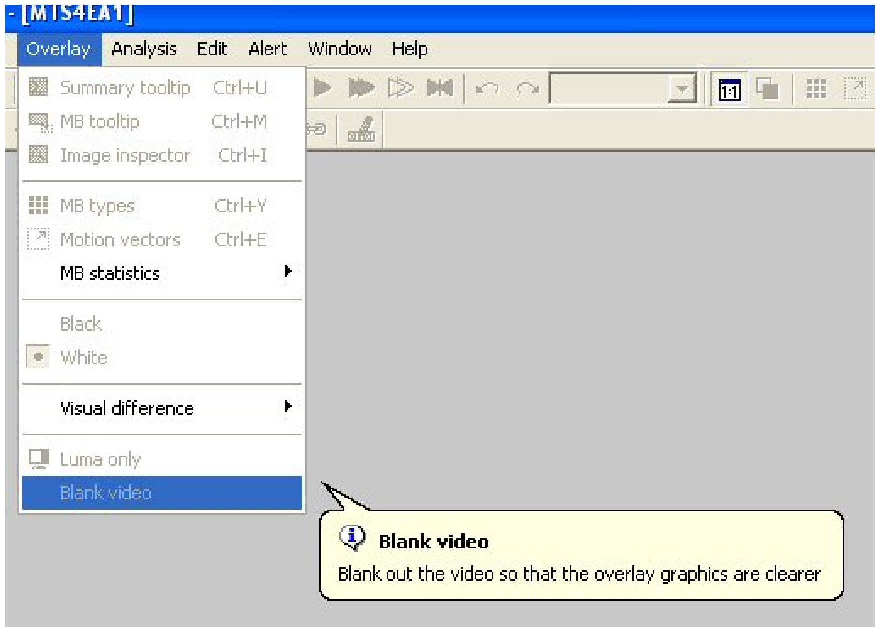


The gray background in the window indicates that no video or other files are loaded. The only icons on the toolbar which are currently enabled are the icons:

- For the Project (New Project, Open Project and Save Project)
- Open stream... and
- 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 pop-up 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:

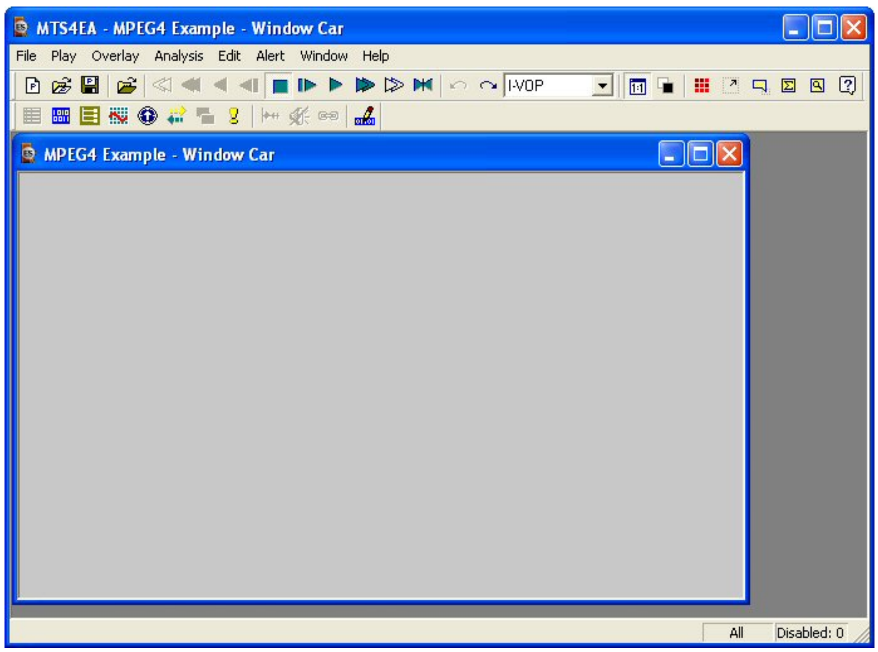


Opening a file

To open a file:

- Click on File, Open stream... or
- Click the Open file icon, or
- Press `Ctrl+O`

Once a file is opened, the display initially looks as follows:



As is normal with Windows applications, the window can be dragged around within the frame of the application, re-sized, 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 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 or Fast forward or Blind 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 **Trace** enable `Ctrl+T` on page 6–112 and **Graph** enable `Ctrl+G` on page 6–129 for an explanation)
- The real-time onscreen image overlays, such as MB types (see *MacroBlock* Types `Ctrl+Y` on page 6–86) and motion vector overlay (see **Motion** vectors `Ctrl+E` on page 6–91) do not update the video window when in Stop mode

The manual states below where this is the case, in the relevant section.

***NOTE**. See also **Audio-video** synchronization on page 6–43 for further information and details on playing audio and audio-video synchronization.*

MTS4EA projects

When working with MTS4EA, the projects feature allows all current set-ups to be saved at any point and re-loaded, for example, to allow work to re-commence at exactly the same point. The project file includes information on the name of the file being analyzed, the views open and the location of these, plus many other items. See **New** project... on page *6–32* for more information.

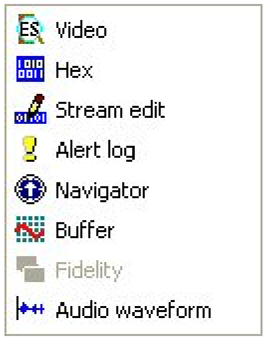
Synchronized views/navigating the views

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

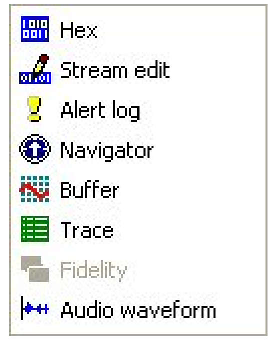
- Video view
- HexView
- Alert log
- Trace views
- Buffer analysis view
- Video navigator view
- Fidelity view (PSNR, etc. analysis)
- Audio waveform view
- File structure view
- Excel graphs

The first eight 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 view, to make it easier to see which area of the video window is providing the link to another view, there is always a small square or rectangle on the video view which highlights the MacroBlock for cross-referencing in the other view.*

Example of moving between views

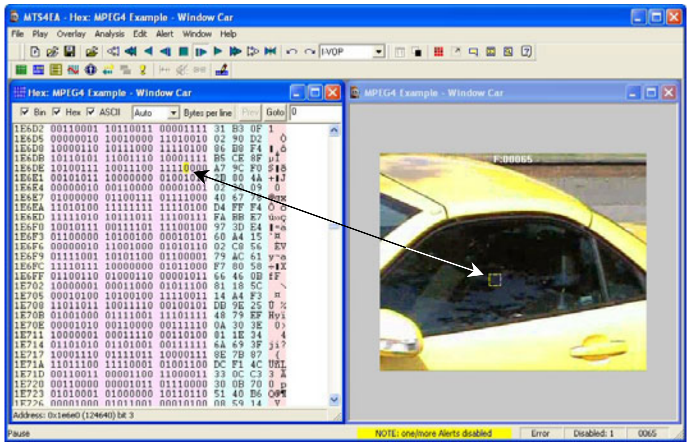
For example, when in the video view and using the MacroBlock tooltip, to see the hex/binary data for a specific MacroBlock, simply right-click at the MacroBlock location of interest and select Goto view from the context menu, then select the HexView. The HexView window is opened (if it is not already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

Likewise, selecting a specific byte in the HexView, 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).

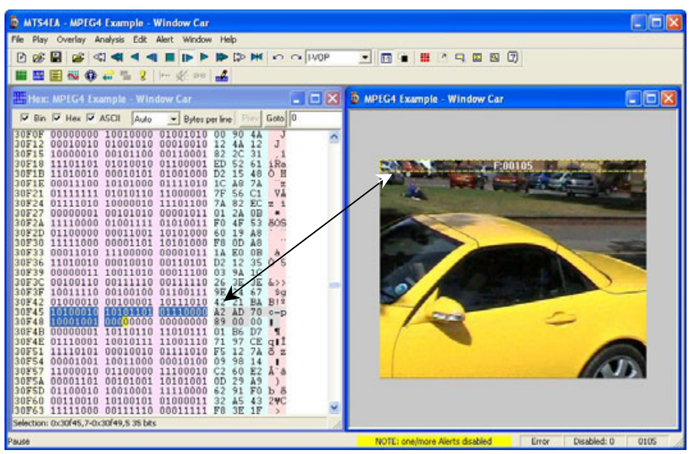
Highlighting of selected areas in the video view

When a data area is selected in a view other than the video view, and then the right-click 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:



- 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 which contains the first selected area:



- 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 which 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.

Synchronization of views/Synchronize views icon

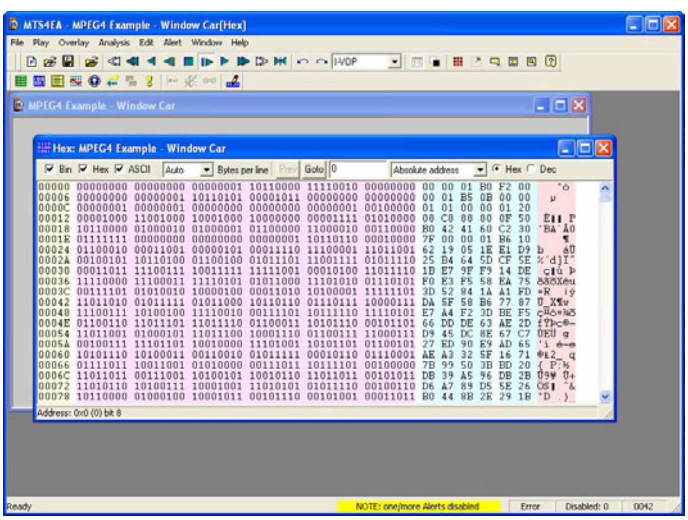
When the Synchronize views icon is pushed in, then all open windows automatically follow the selection that you made. For example, if the video and HexViews are both open, the icon is pushed in, and video is then played, then the highlighting in the HexView window advances so that the highlighted area always corresponds with the relative area in the video window.

***NOTE**. This can cause a delay when playing video for some views (such as the Trace views) or when the memory buffers used by 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 MTS4EA it is possible to open many windows at once. These can be:

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



- Other combinations, such as the video 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 Play, Audio, Audio waveform view... (or by clicking on the icon)

***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 sub-menu options available under each main menu selection, such as File, Play, Overlay, etc., 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 via:

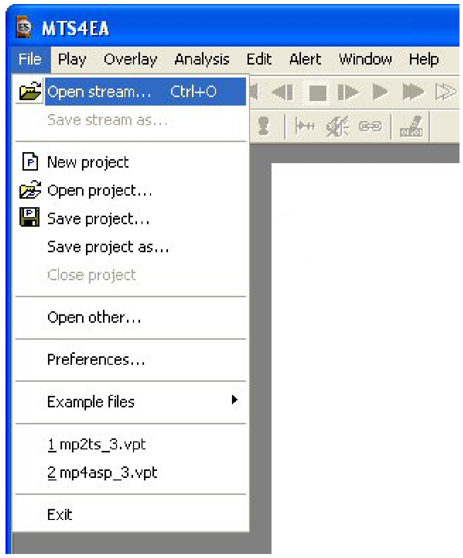
- The icon toolbars (see *Icon Toolbars* on page 6–209)
- Ctrl keys (see **Ctrl** Shortcut Keys on page 6–218)

***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, stop, pause, fast forward, set decoder options
Overlay	Overlay MacroBlock types, motion vectors, data, etc. on top of video
Analysis	Set Trace and Graph options; view Trace, Graph files, Hex view, file structures, buffer analysis, fidelity analysis
Edit	Edit elementary video stream, edit bitstream selection, save edited stream
Alert	Enable/disable/configure overall alert levels and individual alerts
Window	Tile/arrange windows, set video scale, HexView, window list, internal graph view settings
Help	Help topics, PDF help and tutorials, license manager, version info

File Menu



Open stream... `Ctrl+O`

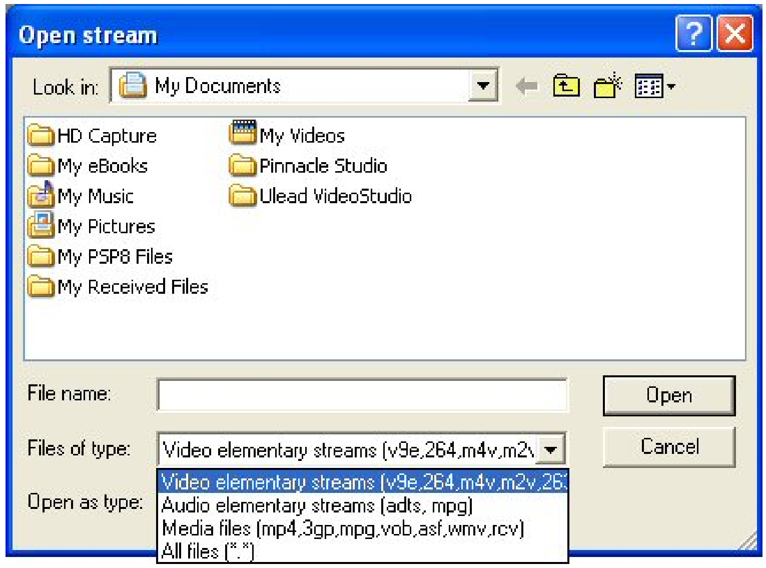
With this option the user is able to browse the file system to find a file to analyze. This can be:

- A file containing only audio, e.g. an MPG file
- A compressed video file which contains video data only, e.g. an MPEG-4 video Elementary Stream (see **MPEG-4** Elementary Stream file format on page 5–17)
- A container file which includes audio and transport information as well as video (see **MP4** files on page 5–12, **3GPP** files on page *5–12*, and *MPEG-2* higher syntactic structures on page 5–13)
- An uncompressed video file (see *Opening an* uncompressed video file (any file extension) on page 6–25

Autodetect file type

When Autodetect is selected in the Open as type drop-down box (as shown in the screenshot above), then the file contents are automatically selected by MTS4EA.

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



The default file extensions that MTS4EA looks for are:

- For audio-only files: `.adts`, `.mpg`
- For video-only files: `.v9e`, `.264`, `.m4v`, `.m2v`, `.263`, `.261`, `.cmp`, `.bits`
- For container files including video: `.mp4`, `.3gp`, `.mpg`, `.vob`, `.asf`, `.wmv`, `.rcv`

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

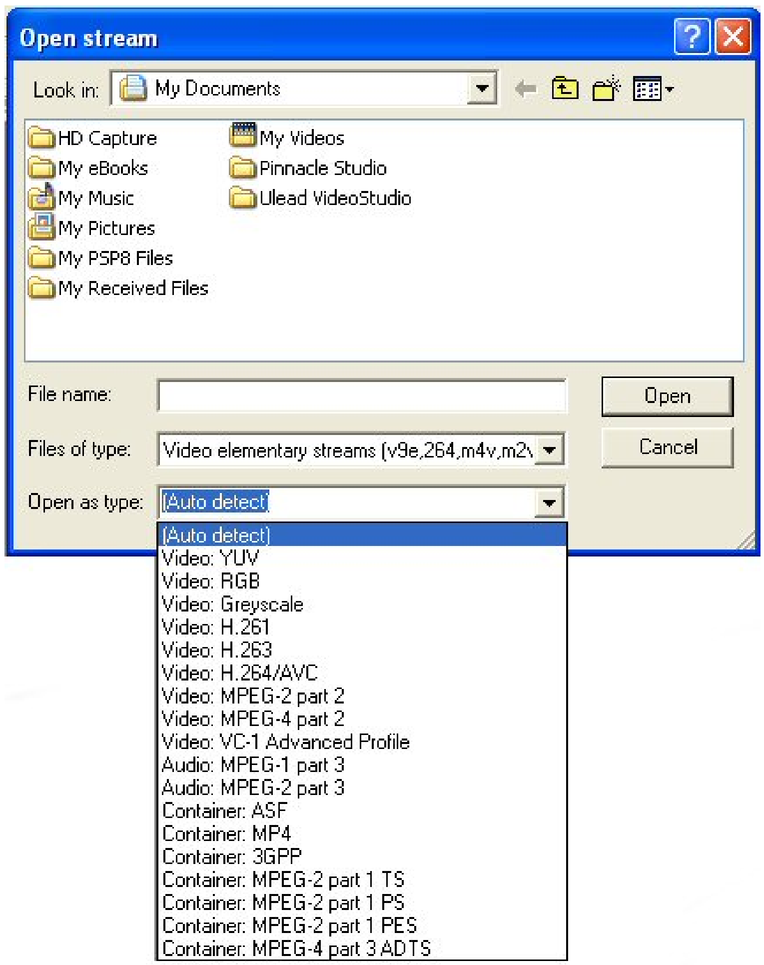
***NOTE**. The exception to this are VOB files, where MTS4EA will take different actions due to the VOB extension (see* Open stream... Ctrl+O *on page 6–13).*

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

Force to open as a specific file type

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

In this case, the file type can be forced from the drop-down Open as type list:



In this case, the file extension is ignored.

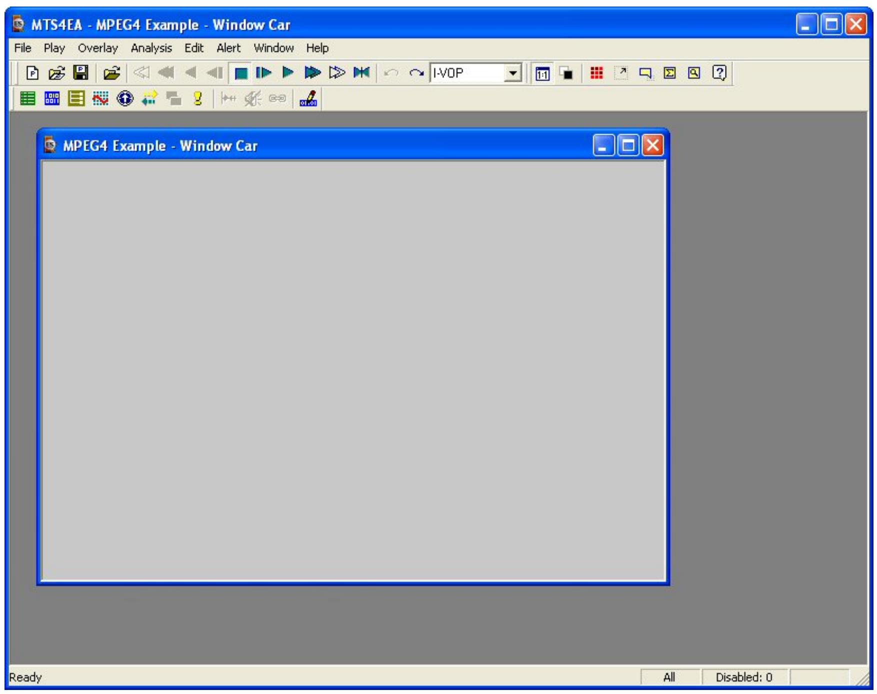
***NOTE**. To open an uncompressed video file, select the appropriate item on the list and MTS4EA will open the file in the appropriate format, regardless of the file extension or the data found in the file.*

Opening a video Elementary Stream (`.m4v`, `.264`, `.m2v`, etc.)

When a raw video elementary stream file is opened, then 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.

Assuming it is possible for MTS4EA to process this standard, then the display appears like this:



The title of the elementary stream is displayed in the title bar of the video window (in this case MPEG4 Example - Window Car, one of the example files provided).

Opening an audio Elementary Stream (`.adts`, `.mpg`, etc.)

When a raw audio elementary stream file is opened, then 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.

Also, because MTS4EA has recognized that the open file is an audio stream, the following icons are now made available in the toolbar:

- Opens the Audio waveform view (see *Audio* waveform view on page 6–60)
- Turns the sound off/on: the stream continues to play and decode even when sound is disabled (see **Mute** audio on page 6–64)

As the file contains audio data only, all other icons for video analysis tools, such as Hexview, Trace, Buffer analysis, etc., are grayed out. The Synchronize audio icon is also unavailable, as there is no video with which to synchronize (see **Audio-video** synchronization on page 6–43).

Opening a container file (`.mp4`, `.3gp`, `.mpg`, `.asf`)

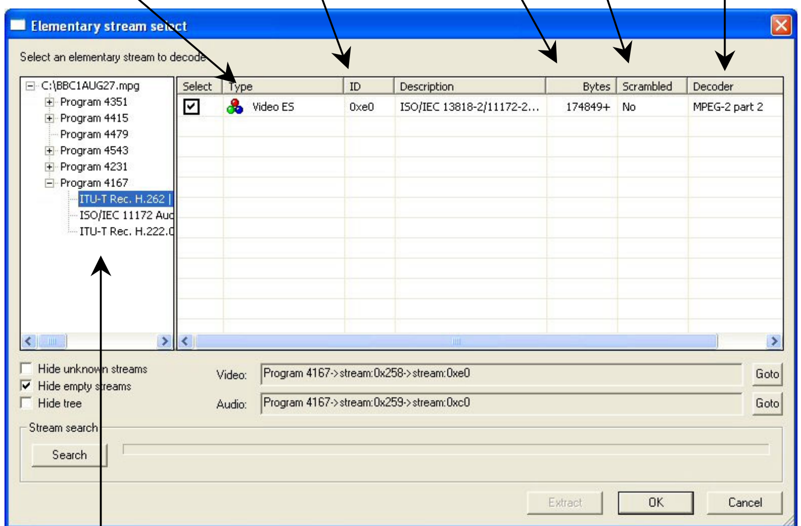
Track type (video, audio, other – note the icon)

Track/stream ID no.

Size of track/stream

Scrambled = encrypted

Decoder type (see below)

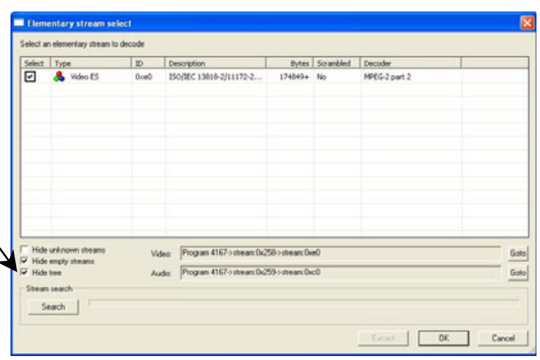


Tree structure showing the various programs in the file (click the + and – to open and close the items)

Streams selected for analysis – enter tick in Select column

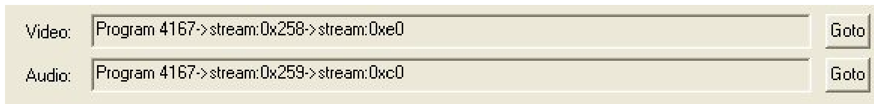
Streams selected for analysis – enter tick in Select column

Tick box below to hide tree structure



When a container file is loaded, 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 the user's choice of decoder, and either immediately played and analyzed within MTS4EA or extracted and saved to a file before going on to play/analyze it.

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



Clicking on Goto will move the view in the browser window to the location in the tree structure containing that stream, where it may be deselected or Decoder 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.

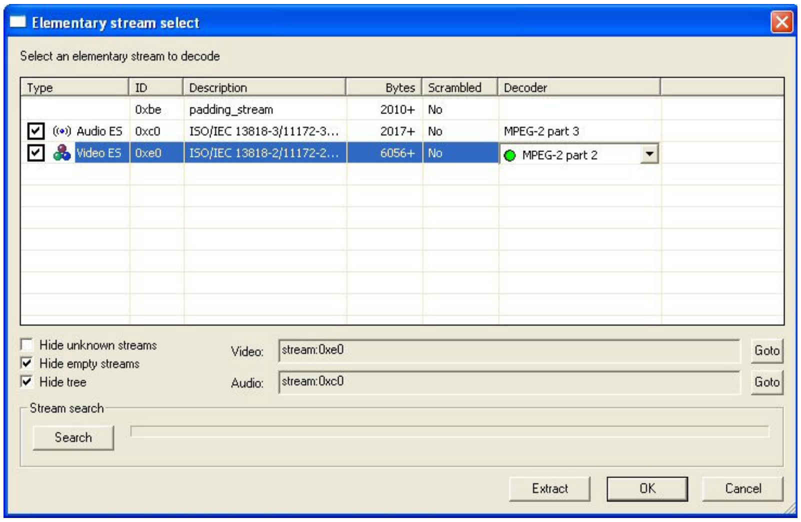
ID. This is the media-stream identifier as dictated by the container signaling information, using appropriate notation for the container standard.

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

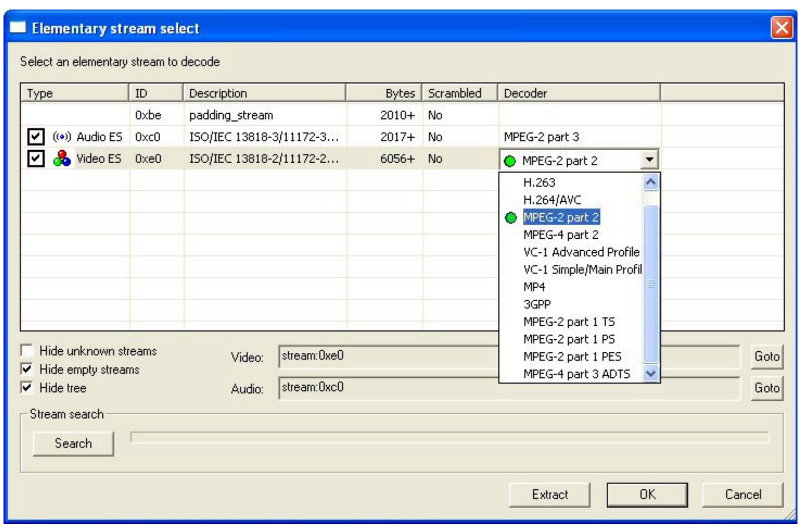
Bytes. This is the number of bytes that will be extracted for the stream. If the value is suffixed by a + sign, the total stream size is not known; without the + sign, this value is the final total.

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 stream concerned. There can be multiple streams containing video or audio, each with different decoders. If the Decoder column is empty, then MTS4EA has been unable to match one of its decoders with the data it found.



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

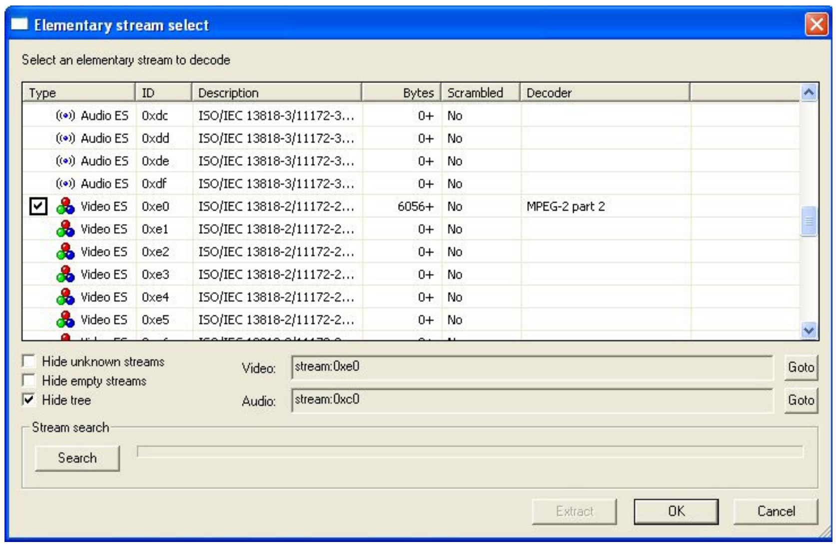


Selecting a different decoder forces MTS4EA to use that decoder for this track/Elementary Stream.

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

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

Hide empty streams. By default, this is enabled. When enabled, this hides streams which 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).



(MPEG-2 example given)

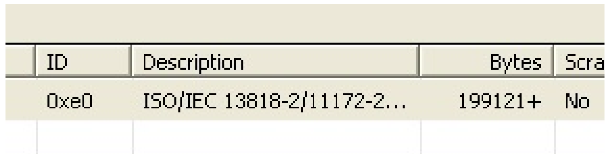
Hide tree. See above.

Search (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, MTS4EA reads the first part of the file only, until a MTS4EA decoder can be matched with the data found.

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

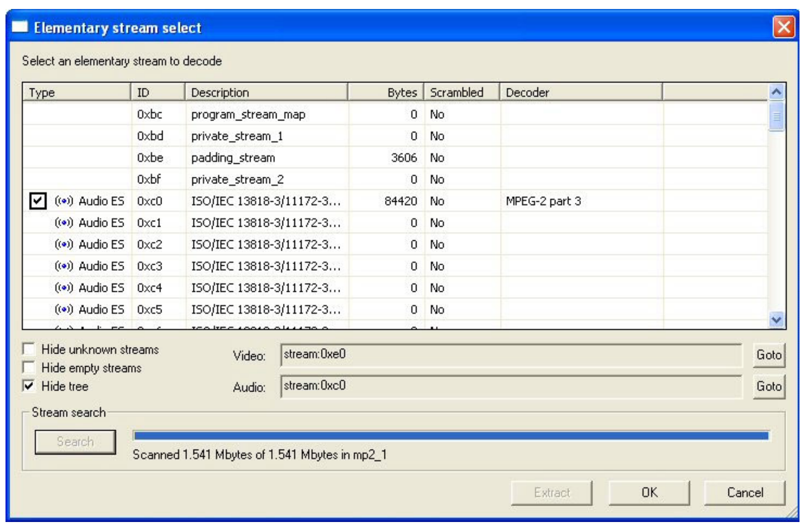


The + sign indicates that the whole file has not been searched and that there may be additional data.

***NOTE**. This 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` *stream hex...* `Ctrl+H` *on page 6–151 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.

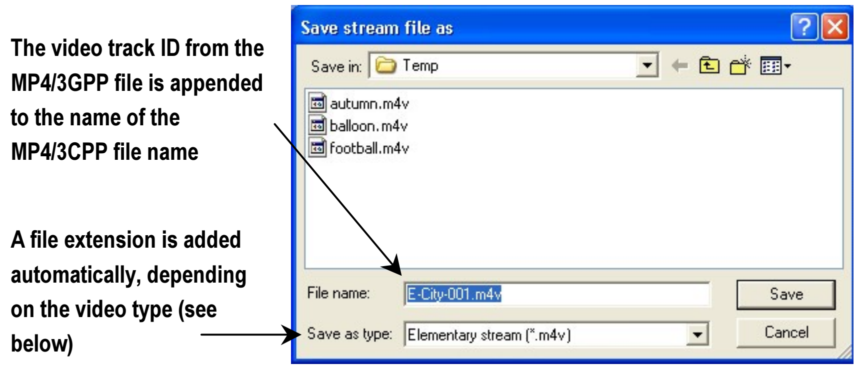
When the search is complete, the display will be similar to that below:



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

Note that during the search process, MTS4EA will search through all the linked files, and therefore may find other streams, and will be able to determine the total size of the streams found: this is why there are now two further streams shown and the sizes no longer have + signs by them (the sizes are now known exactly).

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):



***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** stream as... on page 6–31).*

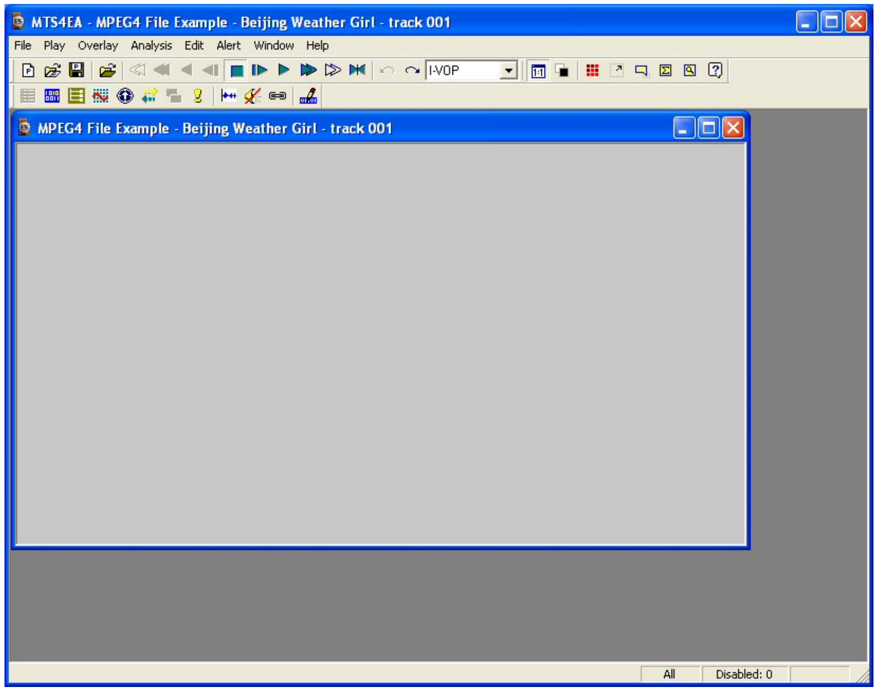
The correct type is automatically added as a file extension, depending upon the data in the file:

- H.264/AVC video format files are given the extension: `.264`
- VC-1 Advanced Profile video files, the extension: `.v9e`
- MPEG-4 video format files, 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`
- H.261 files are given the extension: `.261`

***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 stream selected cannot be opened.

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



Because MTS4EA has recognized that the open file contains both audio and video streams, the Synchronize audio icon, , is now made available in the toolbar (see **Synchronize** audio on page 6–64).

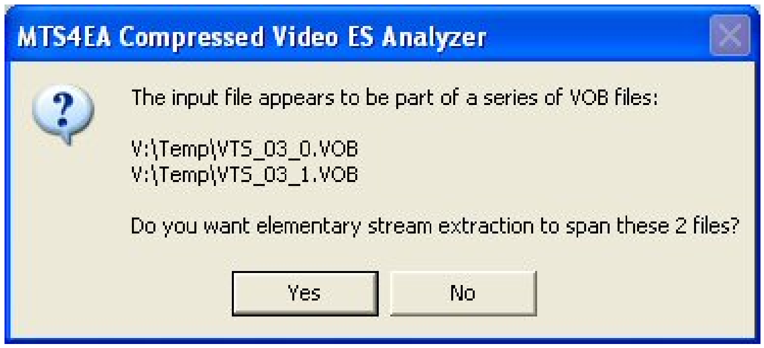
***NOTE**. If you decide to play and analyze the selected stream immediately, and then decide to save it later, this can be done by clicking the Save as... option on the File menu (see **Save** stream as... on page 6–31).*

*In order to examine the structure of the MP4, 3GPP, ASF, MPG (MPEG-2), etc. file, click on the View file structure... option on the Analysis menu (see **View** file structure... Ctrl+R on page 6–158).*

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

***NOTE**. This only applies to files with a VOB extension (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 pop-up box:



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

Once this is done, MTS4EA then uses the standard stream selection dialog box as used for all container files (see *Opening a container file* (.mp4, `.3gp, .mpg, .asf)` on page 6–18).

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 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 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 used by the Microsoft MPEG-4 Part 2 reference encoder Reference [7] (see Standards **References** on page 5–20) 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 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:

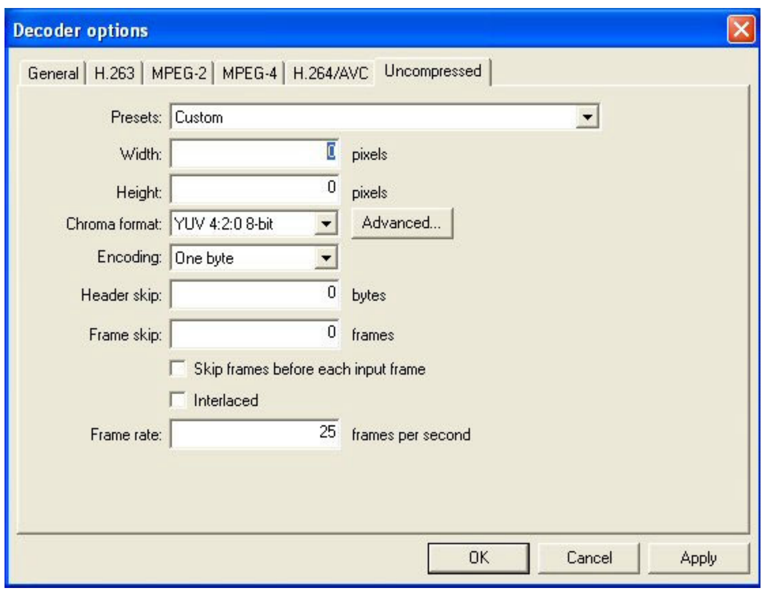
- Concatenated R, G and B planes

For grayscale format:

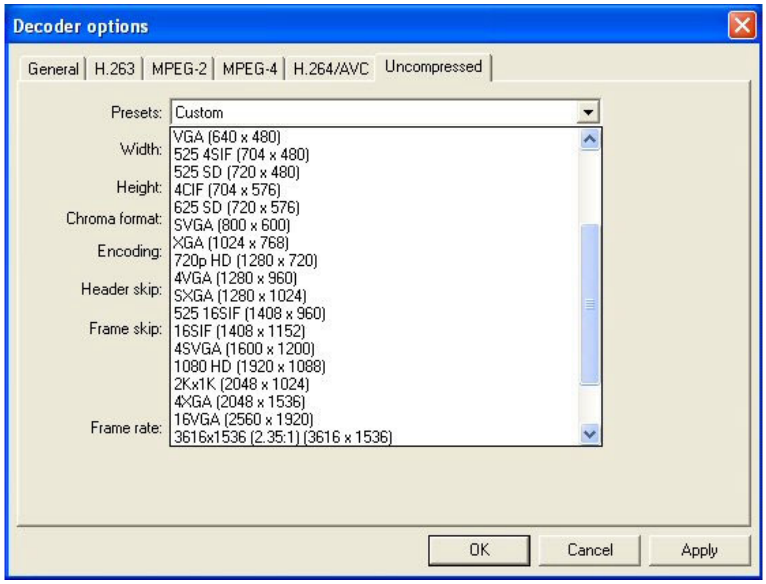
- Luma plane only

When opening an uncompressed video file, by default 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, etc., so when the uncompressed video file is opened, MTS4EA will display the following screen (this is a tab of the Decoder options, which can also be accessed from the Play menu):



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



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* on page 6–30).

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 prior to 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 is skipped before each viewed frame.

Interlaced. Means that 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.

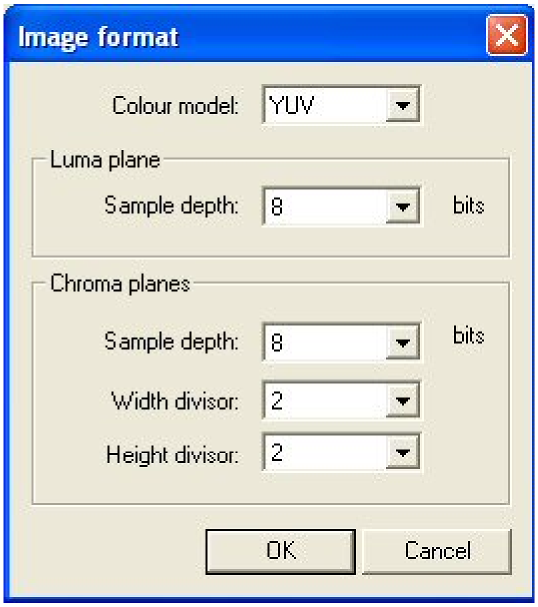
Enabling this check box switches on the Interlace toolbar in MTS4EA which allows the two fields to be viewed independently, either one above the other (top field above bottom field) or either field with the field lines repeated to full frame height.

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 (e.g. 30), or a fraction (e.g. 30000/1001) or a decimal number (e.g. 29.97).*

Image format dialog

The Advanced... button opens the following window:



This dialog 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, or 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.

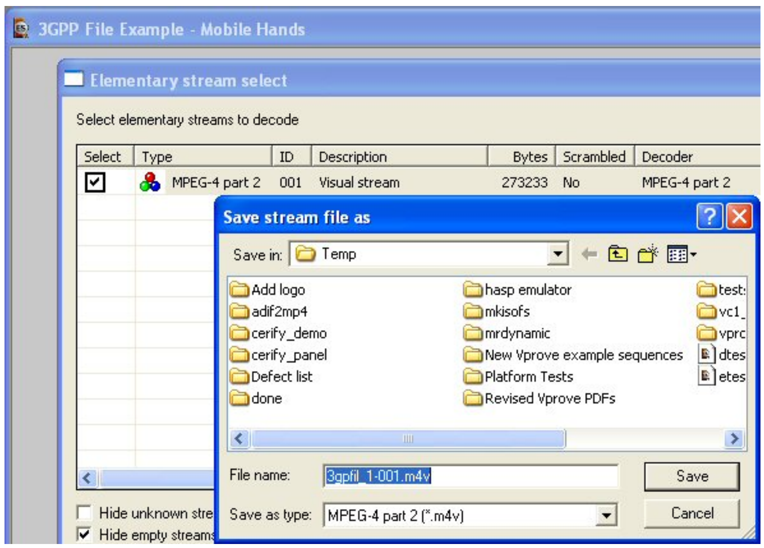
Width divisor. This control specifies the horizontal sub-sampling factor for YUV chroma planes. This option is only available for YUV color models. The value is the factor by which the image width is divided to find the chroma plane width.

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.

Save 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 6–13):



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 (as 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 6–25.

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

New project...

Description of MTS4EA projects

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

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

Project files

The project file has a file extension of `.vpp`. Double-clicking on a VPP file in Windows explorer will open 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, as doing so may prevent 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...

(See also *Description of MTS4EA projects*, **Project** files and **Project** Default file, above, for a description of projects and project files.)

This allows the user to open a previously saved project file.

Save project...

(See also *Description of MTS4EA projects*, **Project** files and **Project** Default file, above, for a description of projects and project files.)

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

Save project as...

(See also *Description of MTS4EA projects*, **Project** files and **Project** Default file, all on page 6–32, for a description of projects and project files.)

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

Close project...

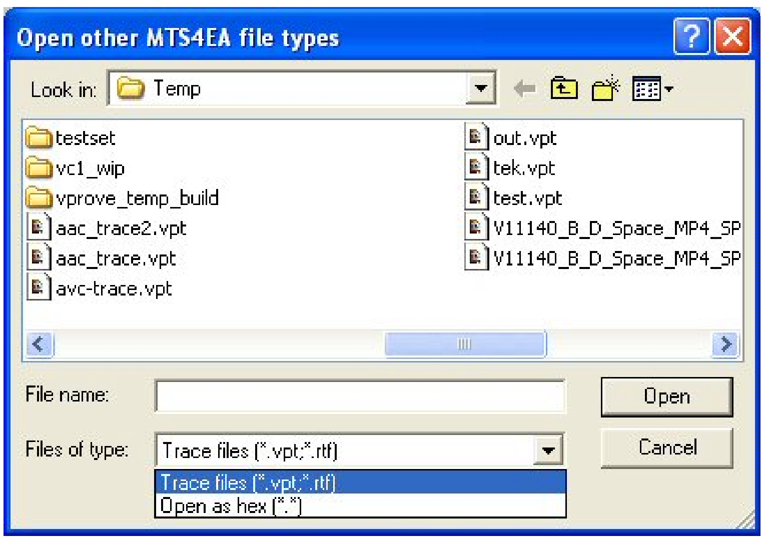
(See also *Description of MTS4EA projects*, **Project** files and **Project** Default file, all on page 6–32, for a description of projects and project files.)

This closes the current project file.

Open other...

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

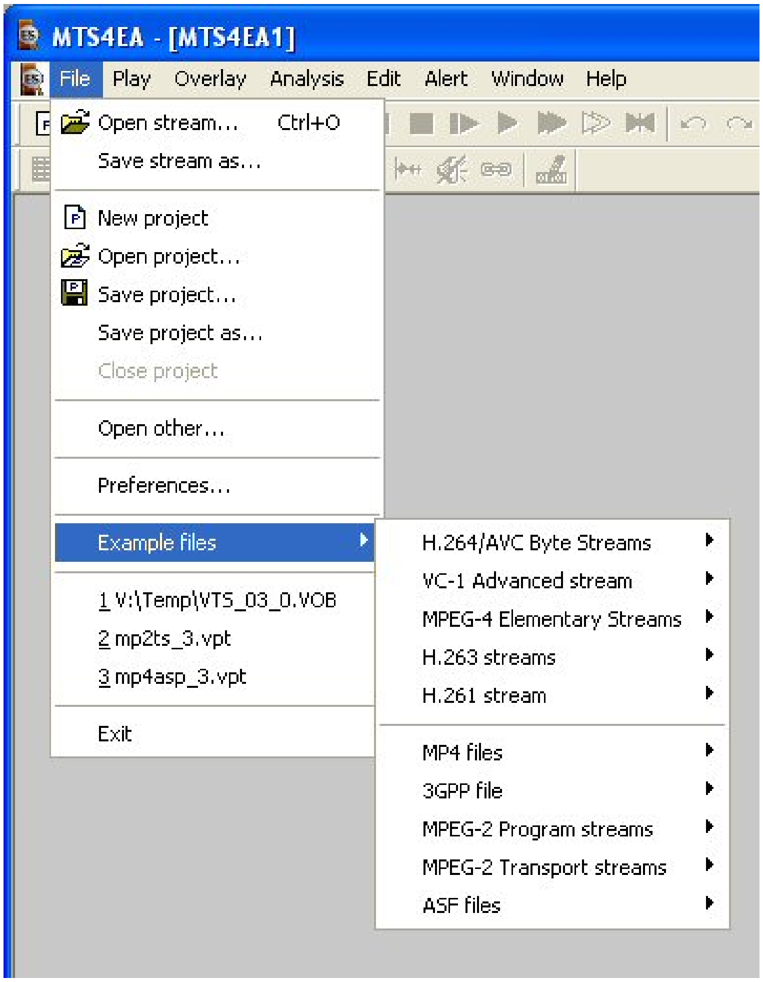
- The MTS4EA trace files, with the file extension `.vpt` or `.rtf`
- Any other file, to open in the HexView



***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 MTS4EA will be `.vpt` files. Currently, `.vpt` files contain only ASCII text but in future they will contain other data also.*

Example files...

Below this option are various example files that can be played/analyzed in 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 pop-up alerts and in the Trace files).

H.264/AVC Byte Streams

These are H.264/AVC compressed video Byte Streams (as described in **H.264/AVC** Byte Stream file format on page 5–15 and Reference [13] under **Standards** References on page 5–20).

Six such 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 *VC-1 Advanced* Profile and Elementary Stream file format on page 5–16 and Reference [17] under **Standards** References on page 5–20). 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 such 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 **MPEG-4** Elementary Stream file format on page 5–17 and Reference [1] under **Standards** References on page 5–20).

Six such 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 **H.263+,** H.263 file format on page 5–18 and Reference [2] under **Standards** References on page 5–20).

Three are provided:

Name	Standard/Annex	Warnings/Errors
Rally (250k)	H.263 baseline	(1) PSUPP is sent when 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 -

H.261 stream

This is an H.261 compressed video file (as described in **H.261** file format on page 5–18 and Reference [3] under **Standards** References on page 5–20).

One such file is provided:

Name	Standard	Warnings/Errors
Conference Room	H.261	Tref (temporal reference) incorrectly set in each frame

MP4 files

These are MPEG-4 `.mp4` container files, containing compressed video and other data (as described in **MP4** files on page 5–12 and Reference [11] under **Standards** References on page 5–20).

Three 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 (audio)	VBV overflow

3GPP file

This is a 3GPP `.3gp` container file, containing compressed video and other data (as described in **3GPP** files on page 5–12 and Reference [12] under **Standards** References on page 5–20).

One such 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 *MPEG-2* higher syntactic structures on page 5–13 and References [14] and [15] under **Standards** References on page 5–20).

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 **MPEG-2** Transport Stream files on page 5–13 and Reference [15] under **Standards** References on page 5–20).

Three 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 -

ASF files

Two Microsoft ASF files are provided, containing compressed video and other data (as described in **Microsoft** ASF files on page 5–12and Reference [18] under **Standards** References on page 5–20).

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

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, as the YUV files are so large.

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)

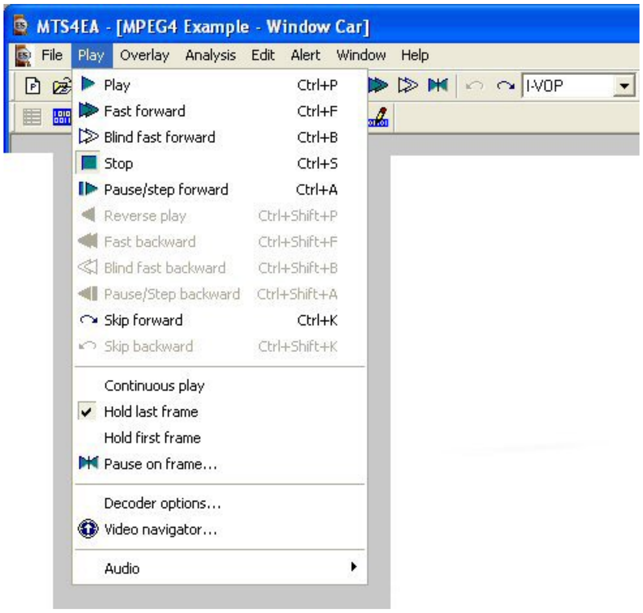
MTS4EA then uses this to determine in which window the file should be re-opened.

Exit

This option exits from MTS4EA.

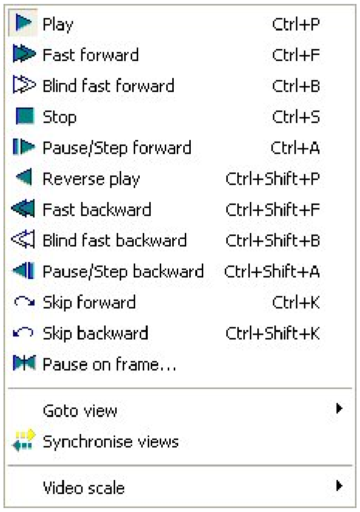
Play Menu

This menu controls the playing of the compressed file:



Right-click pop-up Play menu

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



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 6–203.)

Audio-video synchronization

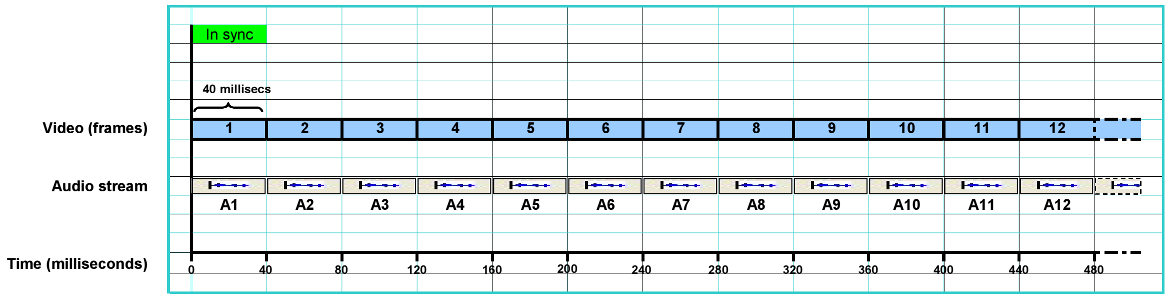
MTS4EA will decode and analyze audio as well as video streams, but the audio will only be played and audible when the stream is in forward play mode.

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 poss. without display		Decodes video file as quickly as poss. 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		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 muted	Skips on to specified frame/type/time/number of frames	Audio muted
	Skips back to specified frame/type/time/number of frames		Skips back to specified frame/type/time/number of frames	Audio muted	Skips back to specified frame/type/time/number of frames	Audio muted
	Pauses when specified video frame is reached		Pauses when specified video frame is reached	Audio muted	Pauses on specified frame	Audio muted

Audio sounds broken up

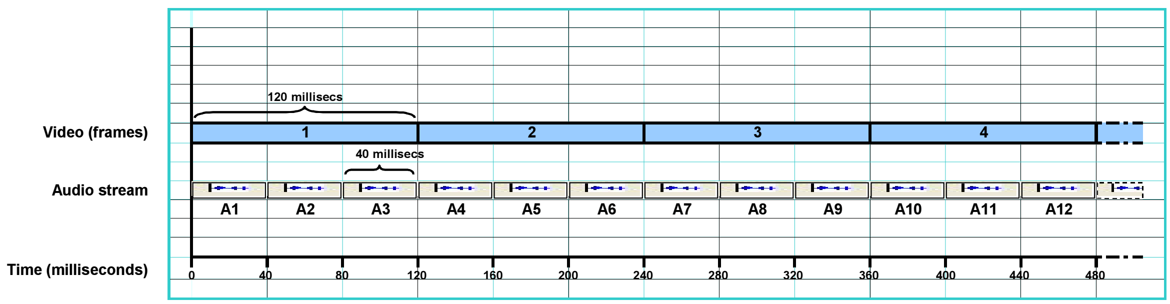
When a file containing both audio and video streams is played and the Audio synchronize function is engaged, if the PC is not sufficiently powerful, it may be that synchronization is not possible and the resultant audio heard by the user will be broken. The figures below describe, in simplified terms,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



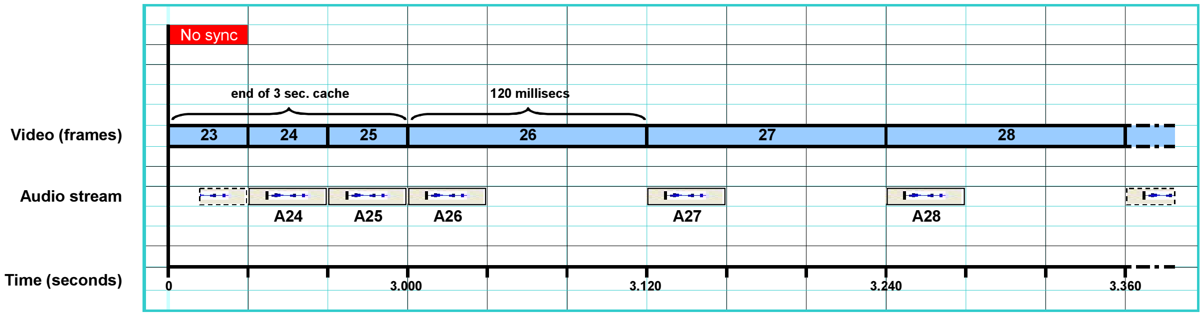
The illustration above shows video being decoded at a display rate of one frame every 40/1000 seconds. Audio plays, unbroken, at the same rate as the video.

When synchronization is not requested, the Synchronize audio icon is not depressed, 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 diagram) as it attempts to keep pace with the video by means of timestamps.

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 quite 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 6–64 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 prior to display of 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 upon 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 fast forwards and the audio is muted.

NOTE. *When playing video forwards or backwards, 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 10-500 frames.*

To change the buffer size, use the Play menu, Decoder options, General tab. See ***Decoder*** *options... on page 6–51 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 on 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... then 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.

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 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` above 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.

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

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

- Hold last frame is selected, in which case the last frame displayed 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 or by pressing `Ctrl+A`, or by clicking the tool bar icon Pause/Step forward .

This command is unavailable when playing audio files. 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

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

***NOTE.** See the Note under **Fast** forward `Ctrl+F` above 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` above.

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.

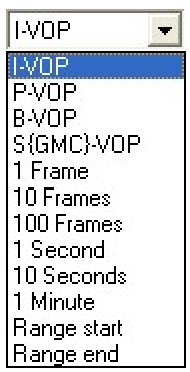
For example, if the box next to the skip forward icon has I-VOP in it (e.g. I-VOP) then clicking the Skip forward icon will skip to the next I-VOP.

This option can be selected from the Play menu or by pressing `Ctrl+K`, or by clicking the tool bar 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 **Play** `Ctrl+P` on page 6–45 and **General**, **Decoder** options... on page 6–51 regarding the buffer used to hold the video and associated data. When skipping forward or backward beyond the end of the buffer, MTS4EA must re-load the buffer, so some delay may be experienced.

The possible selections vary with the video standard concerned; 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 above.

This option can be selected from the Play menu or 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.

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.

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

Hold last frame

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

When this option is selected and the video played to the end, the last decoded frame, and not the gray background screen, will be displayed.

This is selected as on by default when MTS4EA starts.

Hold first frame

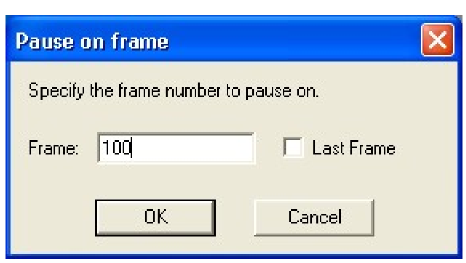
When this option is selected and the video played to the end, the first decoded frame and not the gray background screen, will be displayed.

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

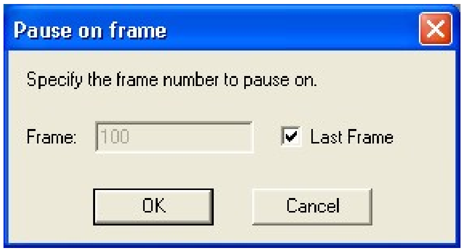
Pause on frame... 

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

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



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 may be enabled again by clearing the Last Frame check box:

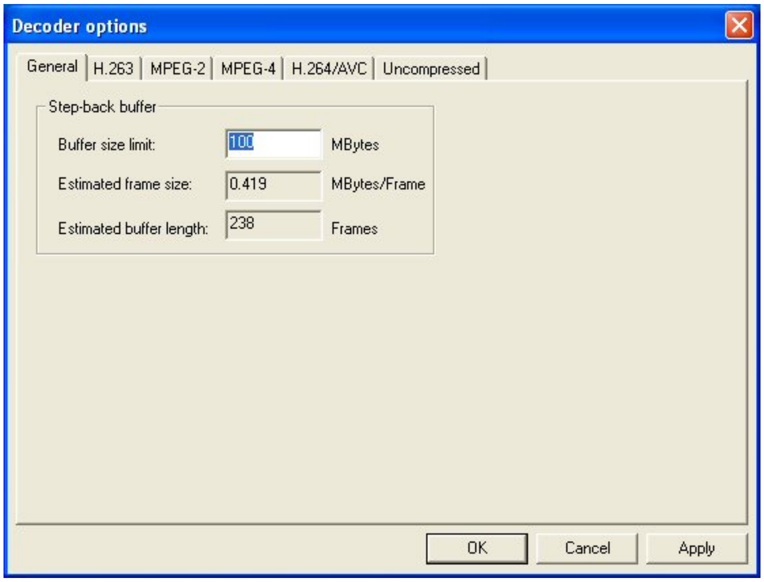


Decoder options...

Below this selection is a dialog box with various tabs that set the options for the video decoder for each standard. When there are no configurable options for a particular video standard then there is no tab on the dialog.

This menu item is unavailable for audio files.

General

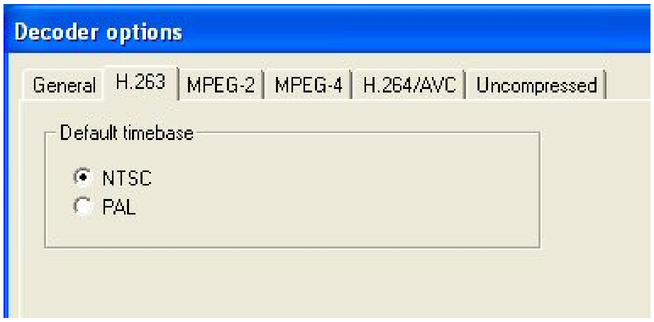


Step-back buffer. This buffer is used to hold the video and associated data, to make video play (forwards and backwards) occur without delay.

***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, 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 10-500 frames.*

The pop-up alerts of Warnings, Errors, etc. which occur when video is decoded by MTS4EA are not displayed if the video and associated data is in the Step-back buffer - see Pop-up alerts when seeking forwards/backwards *through data in the step-back buffer (cached) on page 6–183 for more information.*

H.263

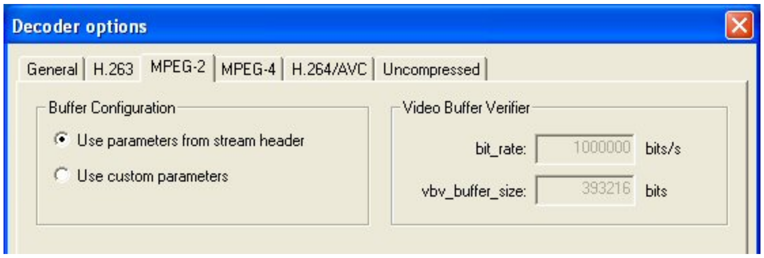


This has no effect and is grayed out for standards other than H.263.

NTSC/PAL option. Within the H.263 standard the NTSC standard is assumed (a frame rate of 29.97 frames per second) and video is played back at this frame rate by default. As a result 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 (at 25 frames per second) does not set these options. In this situation, PAL video would play back at 29.97 frames per second, it would play too quickly. Selecting the PAL option on this tab will play the decoded video at 25 frames per second.

MPEG-2

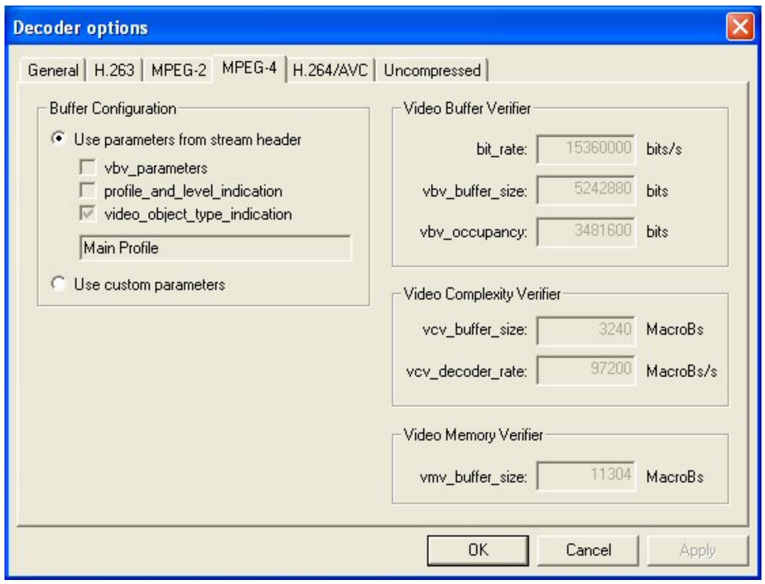


This has no effect and is grayed out for standards other than MPEG-2.

This 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 pages 6–166 and 6–167 for more information.

MPEG-4



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

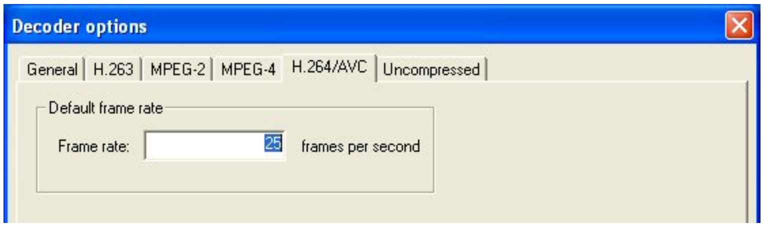
This relates to the use of parameters for buffer analysis (VBV, VCV, VMV) in MPEG-4.

See *Buffer* analysis controls: MPEG-4 and MPEG-2 and Use *custom buffer parameters: MPEG-4 and MPEG-2* on pages 6–166 and 6–167 for more information.

H.264/AVC

Many H.264/AVC byte streams do not provide the SEI 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.



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 Reference [7] under Standards **References** on page 5–20 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.

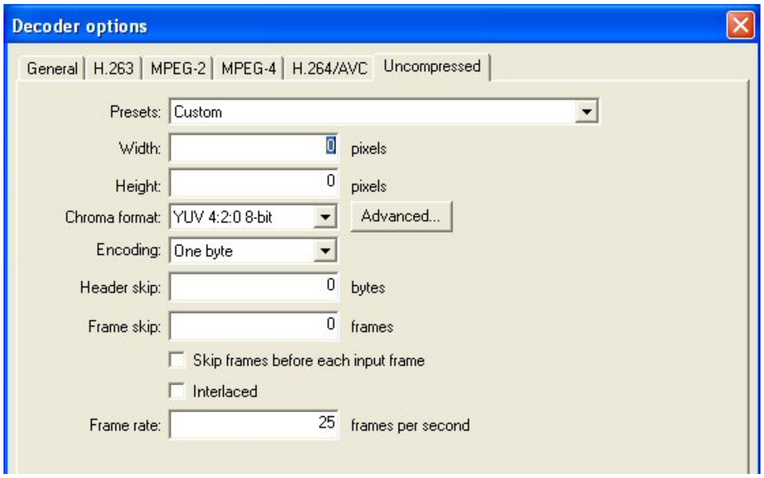
For grayscale format:

- Luma plane only

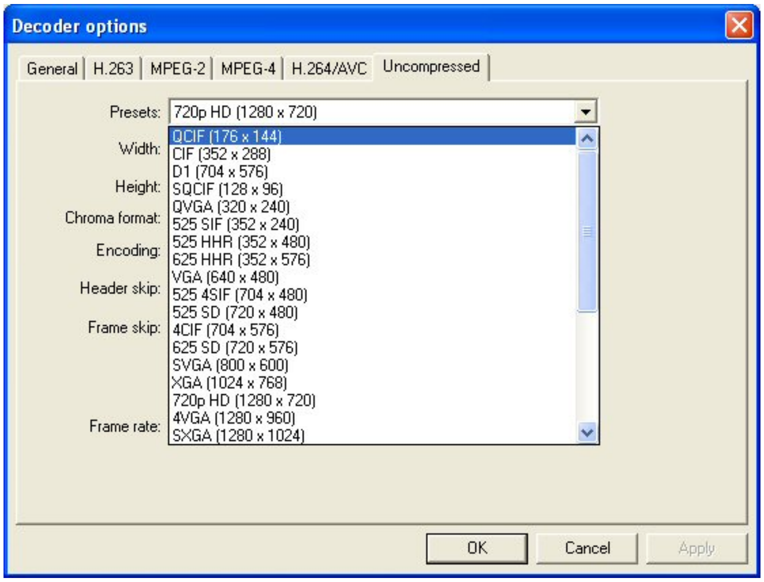
When opening an uncompressed video file, by default MTS4EA shows the uncompressed video decoder options, to enter the required parameters.

NOTE. *Therefore if you try to open an `.m4v` or `.264` or other compressed video file by using open as uncompressed video, MTS4EA will not decode the compressed video but will just assume it is in the format given above.*

Setting the frame size, etc. Within an uncompressed video file, there is no place to indicate the frame size, frame rate, etc., so when the file is opened, 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 (Width and Height). If there are numbers in the filename that could indicate the frame size, then MTS4EA will attempt to read these and offer these in the Width and Height fields shown above. In any event, the correct values can be entered or selected using the drop-down list:



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* on page 6–30.

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 prior to 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 box Skip frames before each input frame, the number of frames are skipped before each viewed frame.

Interlaced. Means that 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 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 (e.g. 30), or a fraction (e.g. 30000/1001) or a decimal number (e.g. 29.97).*

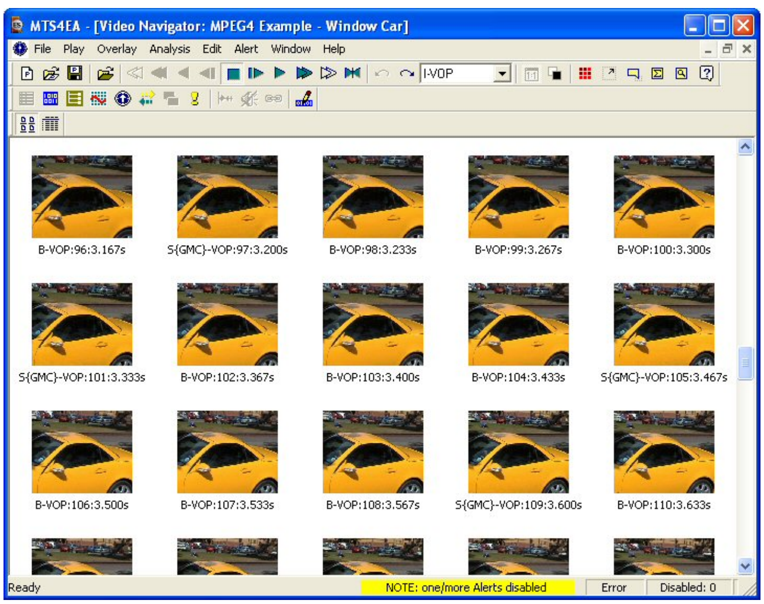
Video navigator...

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** pop-up menu/Goto views on page 6–60.

This menu item is unavailable for audio files.

Thumbnail view



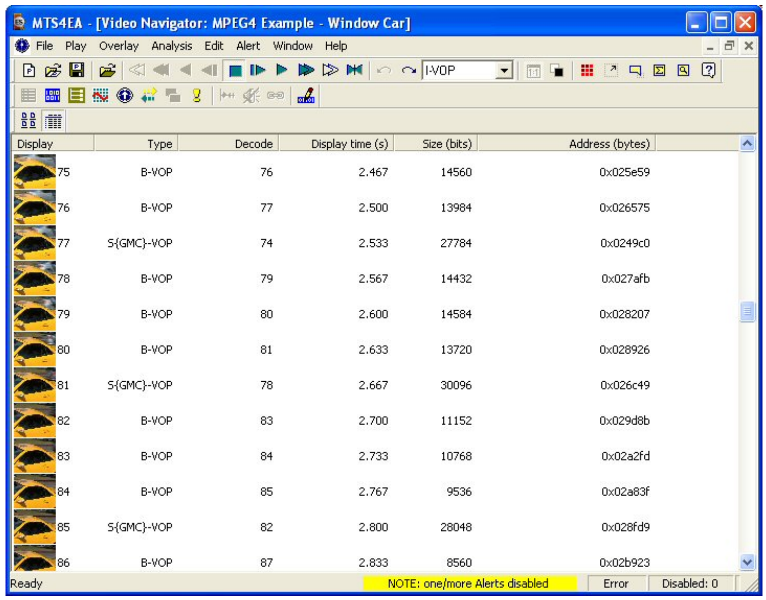
The video frames are shown in display order (decode order can be different if there are B-frames/B-VOPs).

The information given 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, then the frame will be reported as P.

Detail view



The information given is:

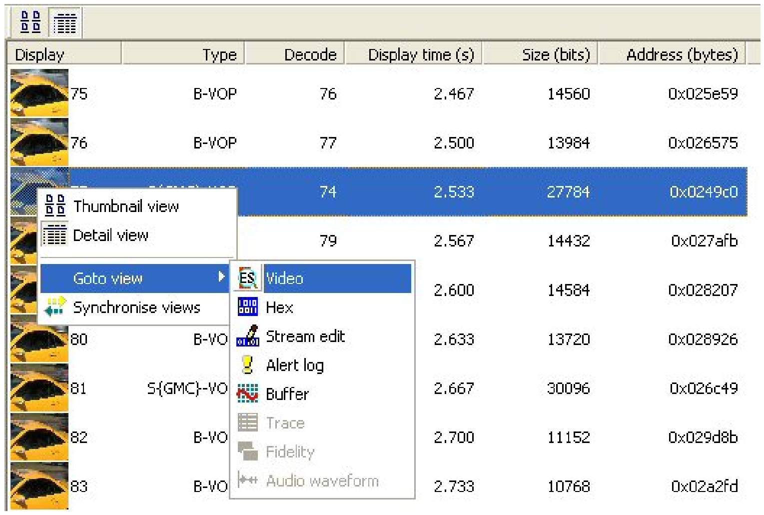
- Display frame/VOP number
- Frame type/VOP type

__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.

- 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 re-sized and dragged into a different order if desired.

Right-click pop-up menu/Goto views



See **Synchronized** views/navigating the views on page 6–7 for more information on navigating between views.

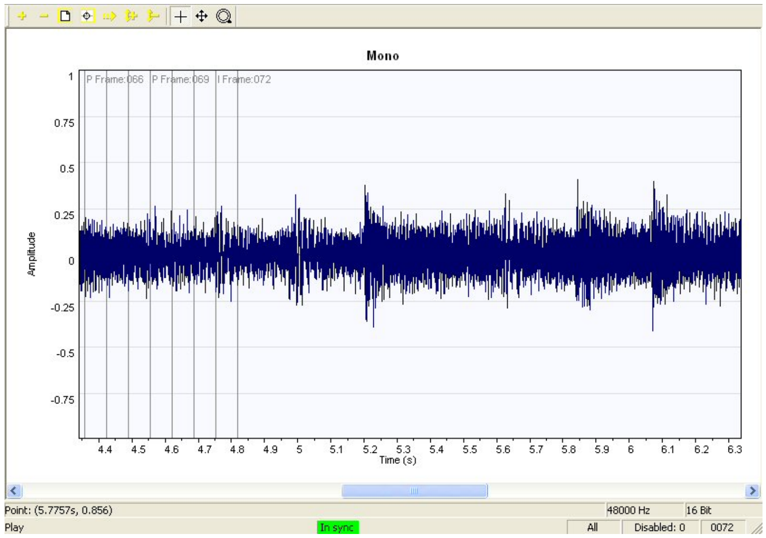
Audio

On the Play menu, Audio has a pop-out sub-section offering three commands. These menu items are only available when a file containing an audio stream or streams is being decoded; at all other times they are grayed out.

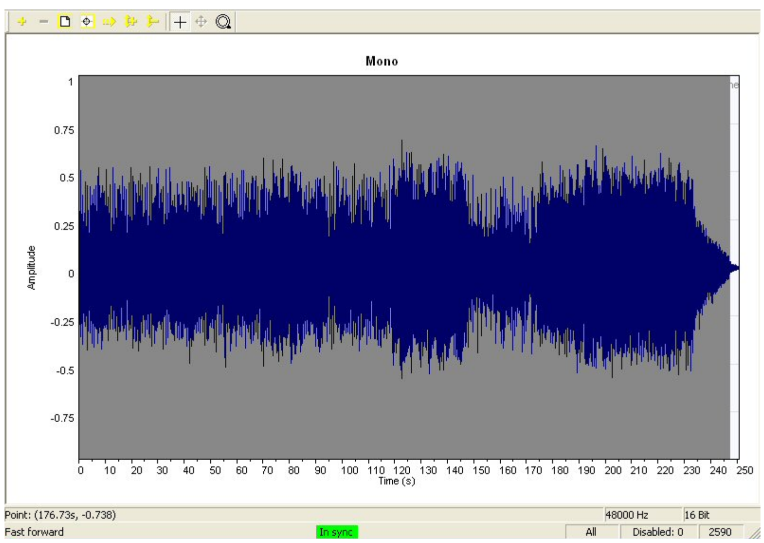
Audio waveform view 

MTS4EA offers the ability to view an audio streams waveform: the y-axis is normalized amplitude, the x-axis is time (seconds).

This can be viewed while the stream is playing, showing the frames being marked off on vertical bars as the stream is decoded:

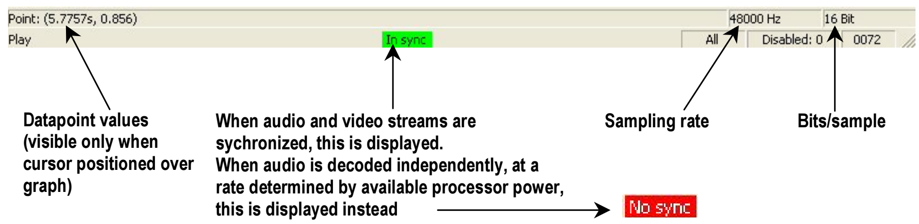


Or when the stream has been played through to its end:

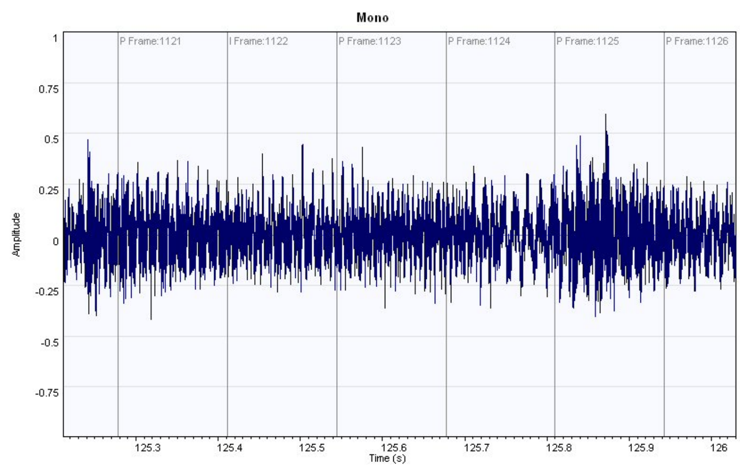


The gray background is due to the density of marked-off frames; when the zoom out icon is used, these will become more spaced and the shape of the waveform more easily discernible (see the following figure). At each frame marker, the frame type and number is shown.

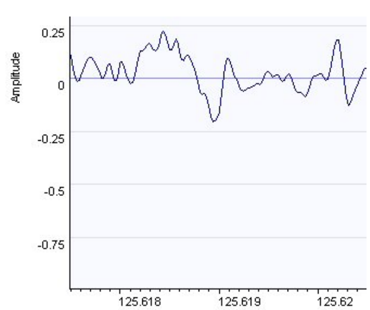
Additional information can be found in the Audio waveform views status bar:

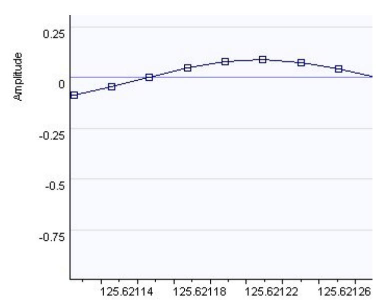


It is possible to zoom in and out on this graph view using the first two icons on the waveform view toolbar, as illustrated below.



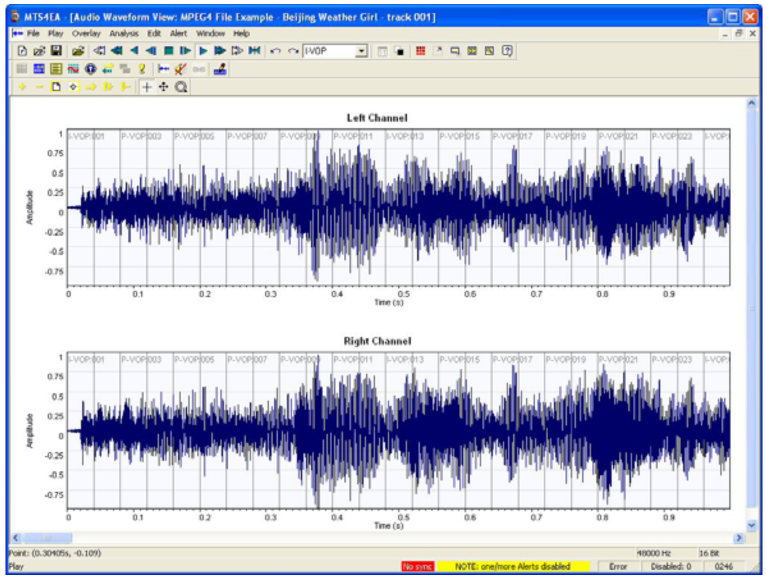
The illustrations below show that it is possible to zoom in so far as to see the individual data points plotted in the audio waveform.



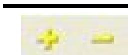
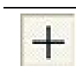


The functions of the other icons are described under **Audio** waveform toolbar icons on page 6–63.

The example below shows the waveform of a stream with multiple channels:



Audio waveform 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
	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

Mute audio 

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

Synchronize audio

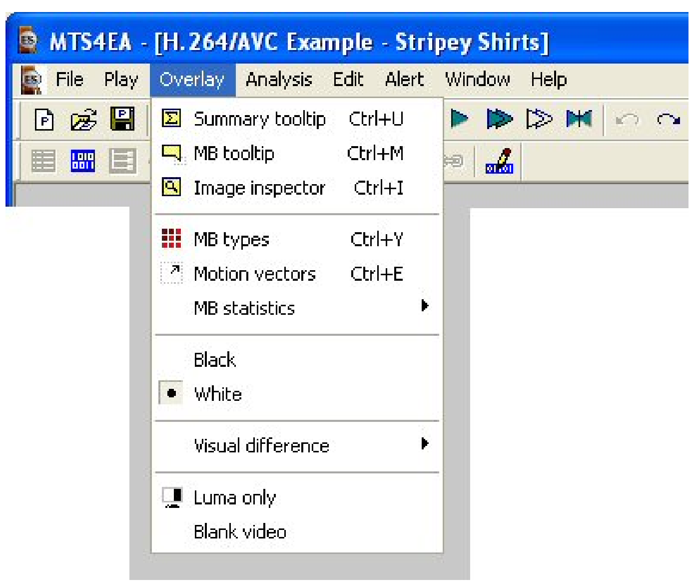
Without synchronization enabled, audio and video streams will play independently, at rates determined by the available processing power of the user's PC, with audio usually playing more quickly than video. When the Synchronize audio icon is depressed, audio data will be played as the video frame to which it pertains is played. Full details can be found under **Audio-video** synchronization on page 6–43.

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

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.

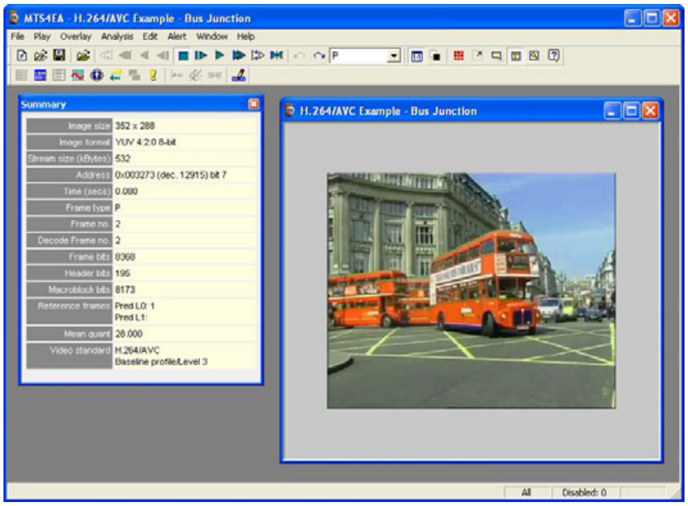
Summary & Macroblock tooltips

Both the Summary tooltip and the MacroBlock tooltip move and dock in the same way.

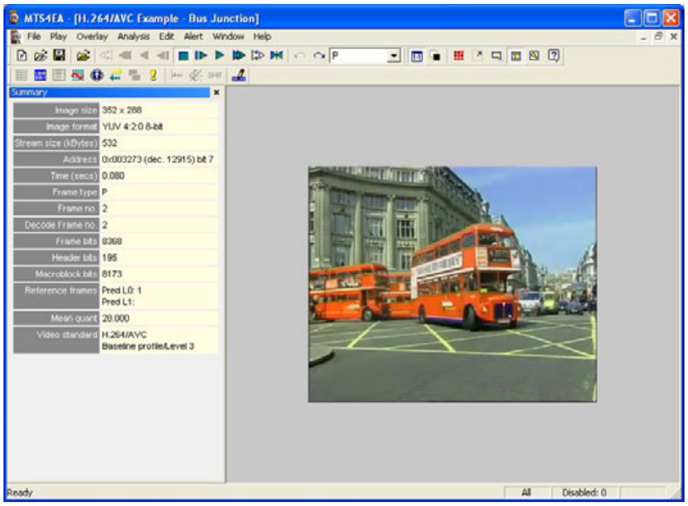
NOTE. *When the tooltips are dragged near to the edge of the window, they automatically dock. To force undocking, press and hold the Ctrl key while dragging with the mouse.*

The tooltips bring up independent windows that can be:

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



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



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

- And the video window (or other 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)

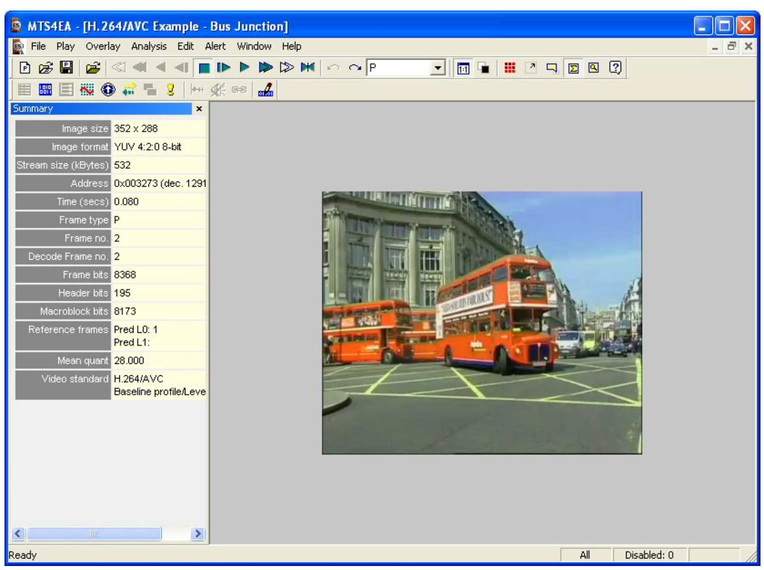
Docking/undocking

- To dock the Summary/MacroBlock tooltip: drag it to one of the edges of the main window (top, left, right, bottom)
- To undock, drag the title bar so that the outline square representing the size of the tooltip appears within the area of the video or other window

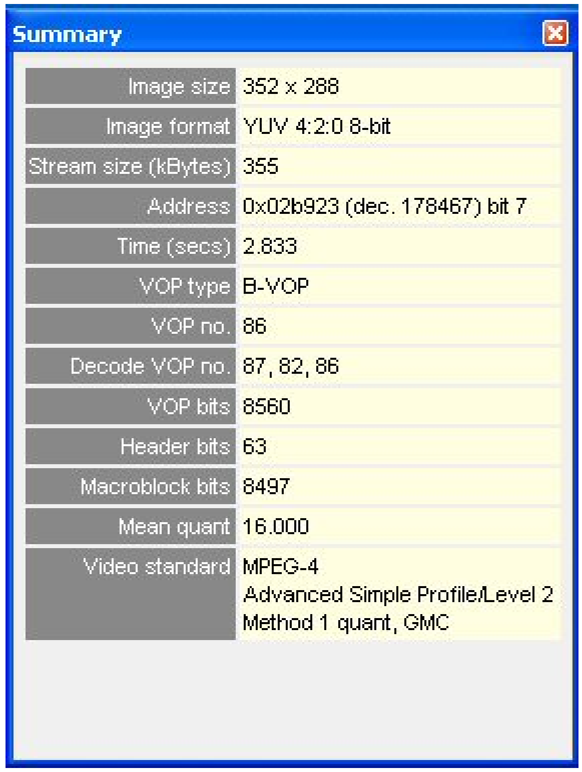
NOTE. *It can sometimes seem hard to undock a tooltip; if necessary, you may need to maximize MTS4EA window to fill the screen in order for there to be a large enough area for the tooltip to undock into. Try dragging the tooltip outline into the center of the video window. It should then change shape to indicate it is no longer docked.*

Scrolling/scroll bars

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



Summary Tooltip Ctrl+U



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 MTS4EA main window.

NOTE. *The titles in the Summary tooltip and the information displayed vary with the video standard concerned 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 and Skip [H.263/H.261 only]

Tref. 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 milliseonds or 40 milliseconds respectively). Tref has a maximum value of 255; when it reaches this value it wraps around back to 0.

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

***NOTE.** Skip and Tref are only displayed for H.263 and H.261.*

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, then 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 in each second.

So, continuing the example of video at 15 frames per second, assuming that there are two Skips before the first frame, then 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

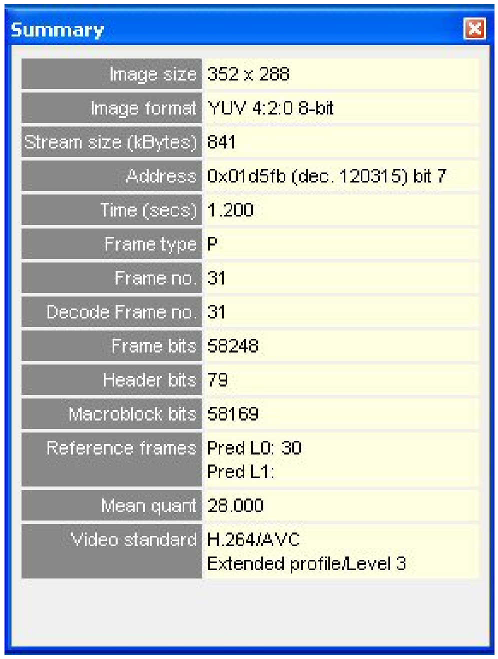
There are many variations to this; often more than one frame is skipped.

***NOTE.** Also, within MTS4EA: for H.263+, H.263 and H.261 video Tref is limited to 8 bits, a maximum value of 255 as this is how Tref is defined within the H.263/H.261 standards (actually TR).*

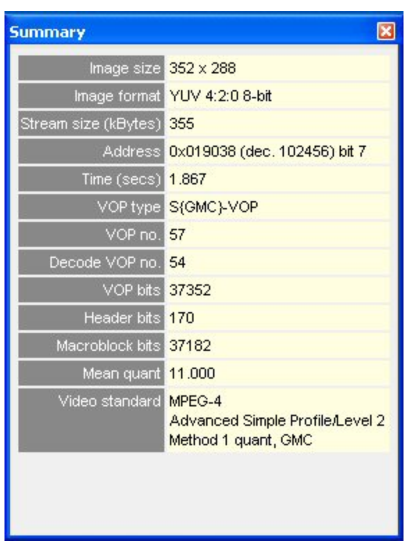
Some of the Trace outputs of MTS4EA keep track of the total number of frames skipped - this is called Cumul. skip (short for Cumulative skip). In the last row of the table in the above example Cumul. skip would be eight.

Summary Tooltip varies by video standard

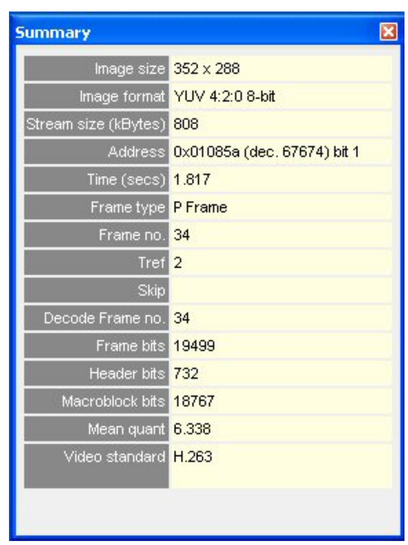
The information provided by the Summary tooltip varies depending upon the video standard that is being analyzed, and the specific options in the video file concerned: the following is an example for an H.264/AVC stream:



And these are examples for:



MPEG-4 Advanced Simple Profile



H.263+

Information provided by Summary Tooltip

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 6–75.*

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, e.g. 0x002681, then in decimal in brackets, e.g. (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 that many H.264/AVC bitstreams do not have the relevant SEI values (as 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, H.261]: the time-stamps in the picture header. If the Temporal references [Tref] are not set or are not correctly set, then 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, H.261, 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 below):
  - 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 then the whole frame is reported as a B-frame; likewise, if any of the slices in the frame are P-slices then 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 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 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, H.261] Frame type: the frame type, I-frame or P-frame or B-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.263/H.261 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, H.261, VC-1]

The number of the displayed VOP [MPEG-4]/frame [H.264/AVC, MPEG-2, H.263, H.261] in the sequence.

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

Tref and **Skip** [H.263, H.261 only]

See under *Tref and Skip [H.263/H.261 only]* on page 6–67.

Decode VOP no. [MPEG-4] or **Decode Frame no.** [H.264/AVC, MPEG-2, H.263, H.261, VC-1]

The data that appears in this field differs with 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 to 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 also 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, H.261, 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, H.261, VC-1] frame

NOTE. *For H.264/AVC, if any of the slices in the frame are interlaced then 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 prior to 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

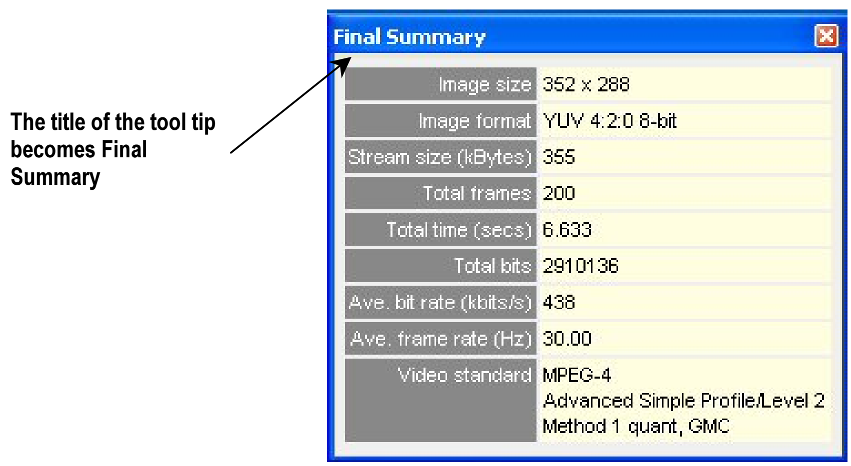
  - [MPEG-4 Advanced Simple Profile] Interlace, QS, Method 1 quant, as appropriate
- [MPEG-2, VC-1]:
  - Profile and Level
- [H.264/AVC]:
  - Profile and Level
  - Entropy coding mode, CAVLC

NOTE. *For H.264/AVC, many of the other elements (e.g. 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 meanings of:

- Image size
- File size (KB)
- Standard

are the same as given under **Information** provided by Summary Tooltip on page 6–69. 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 was playing normally (MTS4EA keeps track of the playing time, regardless if 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.

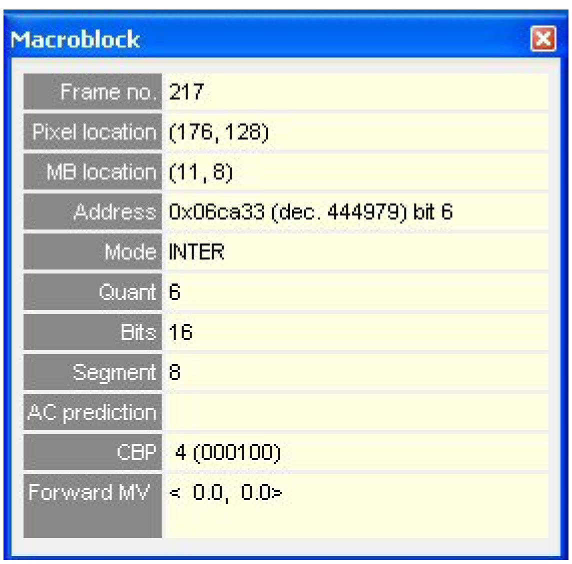
Ave. frame rate (Hz). This is Total frames divided by Total time, in frames/second (Hz).

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`

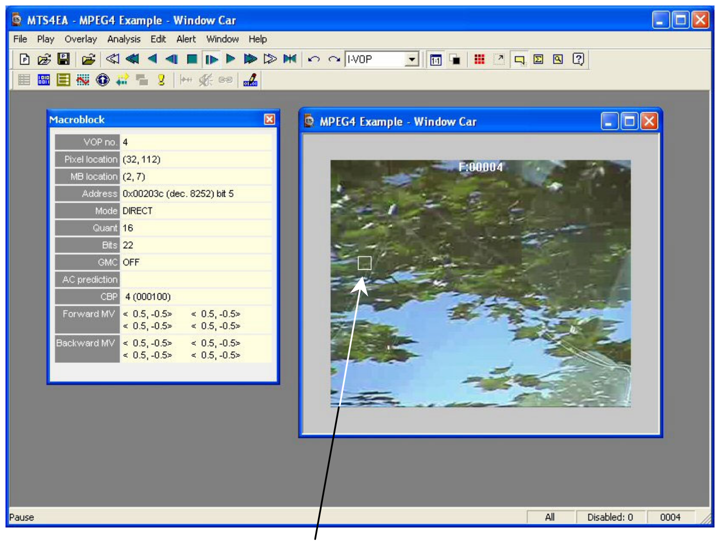
This displays information about a selected MacroBlock:



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 6–77).*

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:



Move the hand cursor to select the MacroBlock

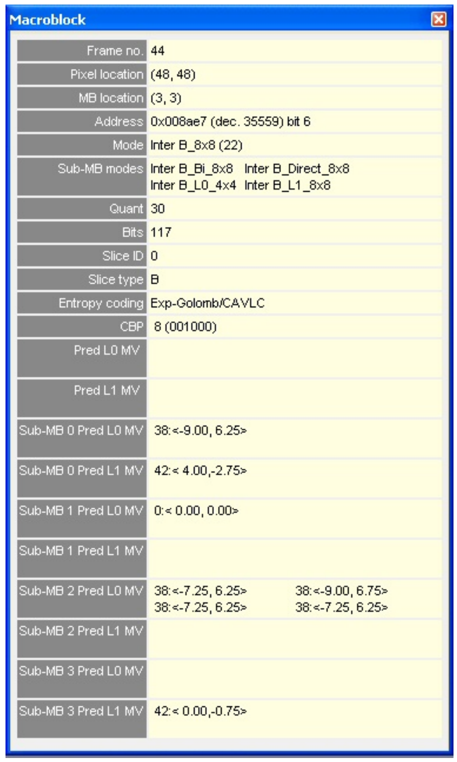
Viewing the MacroBlock data in other views

To see the MacroBlock data in other views - e.g. 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.

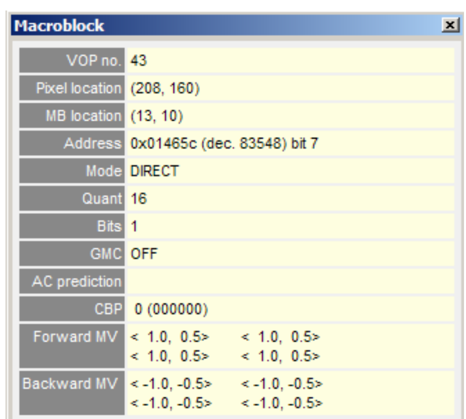
See under **Synchronized** views/navigating the views on page 6–7 for more information.

MacroBlock tooltip by video standard

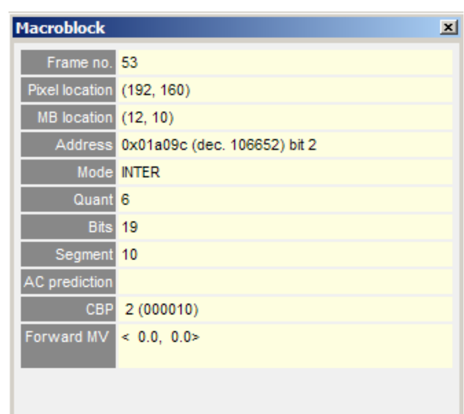
The information provided by the Macroblock tooltip varies depending upon the video standard that is being analyzed, and the options used in the video file:



H.264/AVC



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



H.263 Inter MB (note Segment)

Information provided by MB Tooltip

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, H.261]

The number of the displayed VOP [MPEG-4]/frame [H.264/AVC, MPEG-2, H.263, H.261] 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 **Information** provided by Summary Tooltip on page 6–69 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, e.g. 0x002b8a then in decimal in brackets, e.g. (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 above 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 6–86.

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 6–87.

Quant [All standards]

The Quantizer used by the Luminance (Chrominance may differ) (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, H.261 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, H.263 and H.261]

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 comprise 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, H.263 and H.261 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 forwards 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:

Forward MV (frame/top-field)	In frame coded MBs, shows the forward MVs. In field coded MBs, shows the forward top-field MV
Forward MV (bottom-field)	In field coded MBs, shows the forward bottom-field MV

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 backwards 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:

Backward MV (frame/top-field)	In frame coded MBs, shows the backward MVs. In field coded MBs, shows the backward top-field MV
Backward MV (bottom-field)	In field coded MBs, shows the backward bottom-field MV

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 sub-division of the MacroBlock:

- L0 refers to List 0 reference frames (that is forward reference, from past frames; as an example P type MacroBlocks can only be List 0)
- L1 refers to List 1 reference frames (that 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, e.g.: `38:<9.00,6.25>` means that these motion vectors reference frame 38.

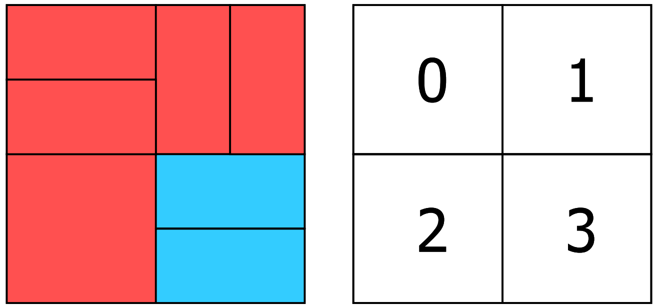
NOTE. *The images of the reference frames can easily be seen using the Video navigator view.*

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 sub-divided (when not sub-divided, there is no data in these fields).

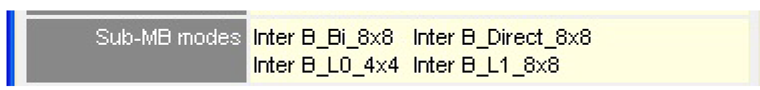
The data provided here follows the standard and is shown in the layout as given in the Sub-MB modes field of the MacroBlock tooltip and MB types overlay.

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.

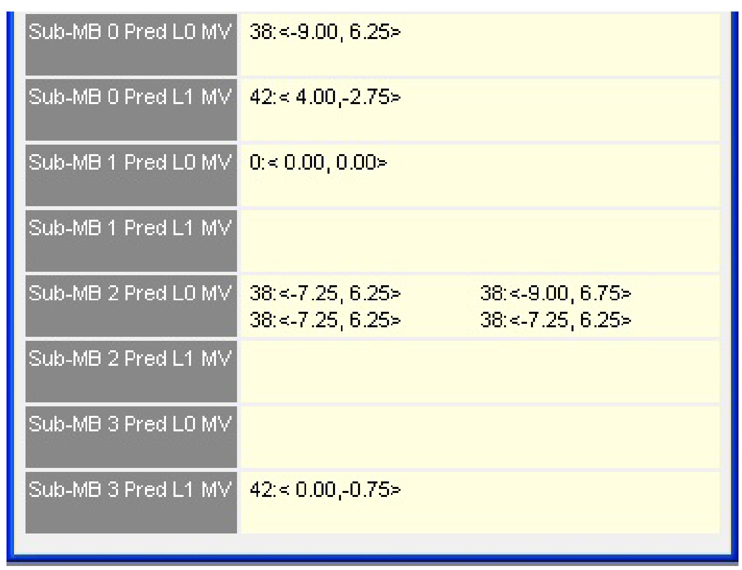
Then the Sub-MB modes field would look like this:



where:

- The top-left reported sub-MB mode is for area 0
- The top-right is for area 1, etc.

and the MV fields would look like this:

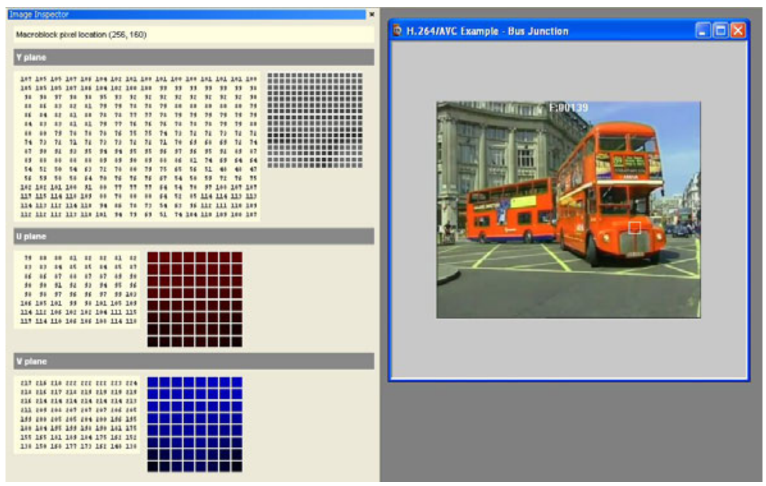


PSNR (Y, U, V) or **PSNR (Y, U, V) (frame/top-field)** and **PSNR (Y, U, V) (bottom field)** [Interlace]

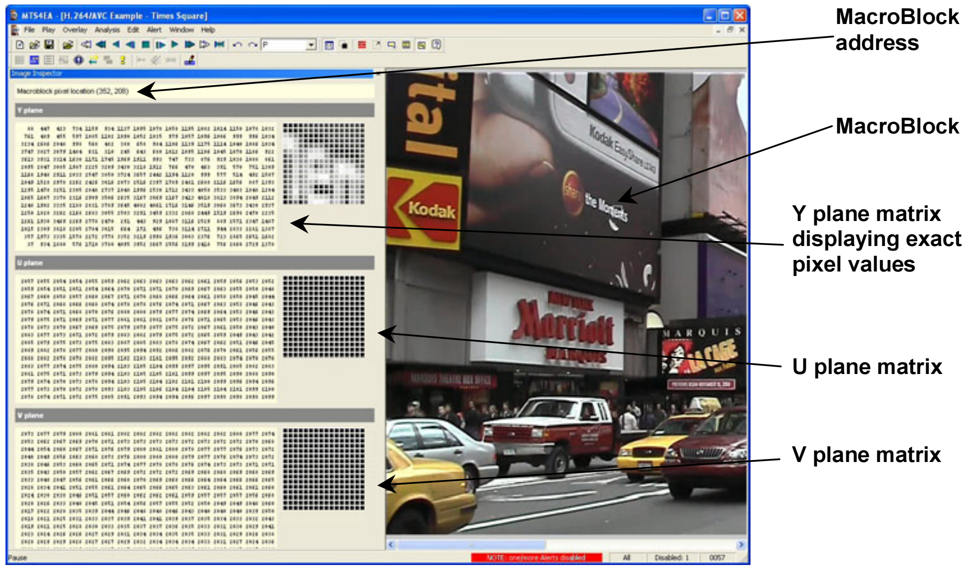
When fidelity analysis is enabled, the selected fidelity analysis value for the selected MacroBlock is displayed.

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).



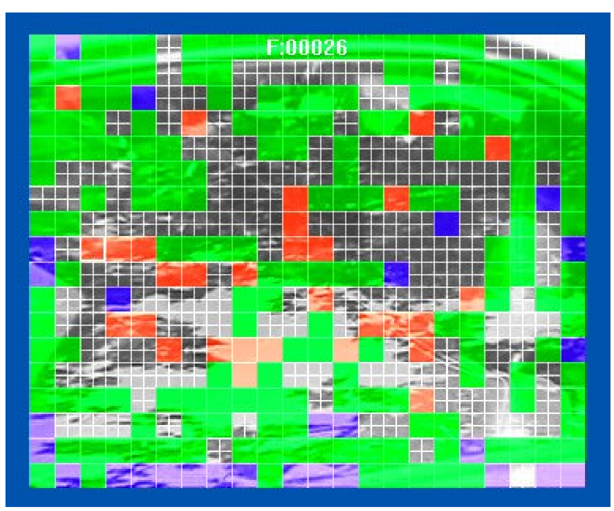
The sub-sampling ratio of these component channels can vary. As can be seen in the example above, when this is the case, the matrices containing the pixel data are sized accordingly. The example above is for 8 bits per sample, 4:2:0 YUV video, the example overleaf for 12 bits per sample, 4:4:4 video.



The colors of the image blocks change, depending upon 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 given above is a B-VOP of an MPEG-4 Advanced Simple Profile stream

This overlay allows the user 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 given below and are also displayed on-screen in the MacroBlock types color-key tooltip (see **MacroBlock** types color key tooltip on page 6–215 for more information).

The lines denote the edges of the MacroBlock, or where the MacroBlock has four motion vectors, then the MacroBlock is also divided into four by the lines. 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] under **Standards** References on page 5–20), 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	I, P, B, SI, SP
I	Intra 4X4	0	-	Yellow	I, P, B, SI, SP
P	Inter list 0	0 - 2	0 - 3	Red	P, SP
B	Inter list 0	1, 4, 5	1, 4, 5, 10	Red	B
B	Inter list 1	2, 6, 7	2, 6, 7, 11	Blue	B
B	Inter list 0 + 1	3, 20, 21	3, 8, 9, 12	Pink	B
B	Inter mixed	8 - 19	-	Light blue	B
B	Inter direct	0	-	Gray	B
SI	Intra 4X4 and 16x16	0	-	Green	SI
SP	Inter list 0	0 - 2	0 - 3	Red	SP

NOTE. *The Intra inferred MacroBlock type was shown in yellow in previous versions of MTS4EA; however, as 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] under **Standards** References on page 5–20), shown in the following colors:

Type	Motion vector	Color	Frame type(s)
Intra	0	Green	I, P, B, BI
Forward	0, 1, 2, or 4	Red	P, B
Backward	0, 1, 2, or 4	Blue	B
Bi-directional	0, 2, or 4	Pink	B

MPEG-4

MPEG-4 has the following possible coded MacroBlock types (see Reference [1] – Table B-1 – under **Standards** References on page 5–20), shown in the following colors:

Index	Type	Dquant/ Dbquant	Motion vector	Color	VOP type(s)
0	Inter		1	Red	P, S-GMC
1	Inter+Q	✓	1	Blue	P, S-GMC
2	Inter4V		4	Pink	P, S-GMC
3	Intra			Green	I, P, S-GMC
4	Intra+Q	✓		Yellow	I, P, S-GMC
-	Direct		8	Gray	B
-	Interpolate	✓	2	Green	B
-	Forward	✓	1	Red	B
-	Backward	✓	1	Blue	B

MPEG-2

MPEG-2 has the following possible coded MacroBlock types (see Reference [14] under **Standards** References on page 5–20), shown in the following colors:

Type	Dquant	Motion vector	Color	Frame type(s)
Intra	✓	1	Green	I, P, B
Forward	✓	2	Red	P, B
Backward	✓	2	Blue	B
Bi-directional	✓	4	Pink	B

H.263+ and H.263

H263 has six possible coded MacroBlock types (see Reference [2] – Table 9/H.263 – under **Standards** References on page 5–20), shown in the following colors:

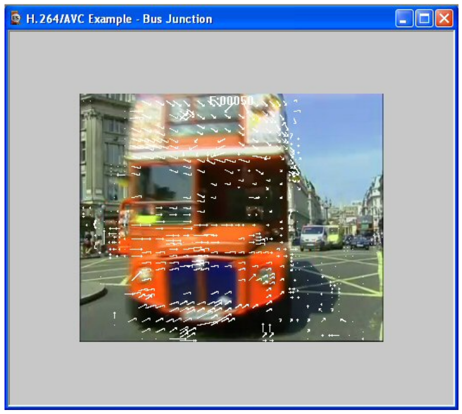
Index	Type	Dquant	Motion vector	Color
0	Inter		1	Red
1	Inter+Q	✓	1	Blue
2	Inter4V		4	Pink
3	Intra			Green
4	Intra+Q	✓		Yellow
5	Inter4V+Q	✓	4	Gray

H.261

H261 has ten possible MacroBlock types (see Reference [3] – Table 2/H.261 – under **Standards** References on page 5–20) shown in the following colors:

Index	Type	Quant	Motion vector	Coeffs	Filter	Color
1	Intra			✓		Green
2	Intra	✓		✓		Yellow
3	Inter			✓		Pink
4	Inter	✓		✓		Gray
5	Inter		✓			Pink
6	Inter		✓	✓		Red
7	Inter	✓	✓	✓		Blue
8	Inter		✓		✓	Pink
9	Inter		✓	✓	✓	Red
10	Inter		✓	✓	✓	Blue

Motion vectors `Ctrl+E`

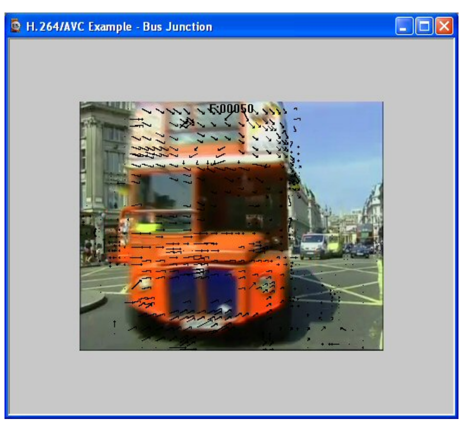


The motion vectors used in relevant video frames can be displayed on the frame currently being viewed. They are switched on and off by doing one of the following:

- From the Overlay menu, select 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 *Black/White* on page 6–110) for all frames except B-VOPs/B-frames - see **Motion** vectors in B-VOPs/B-frames on page 6–92 for information on this.

This is an example of the above video frame with the motion vectors plotted in black:



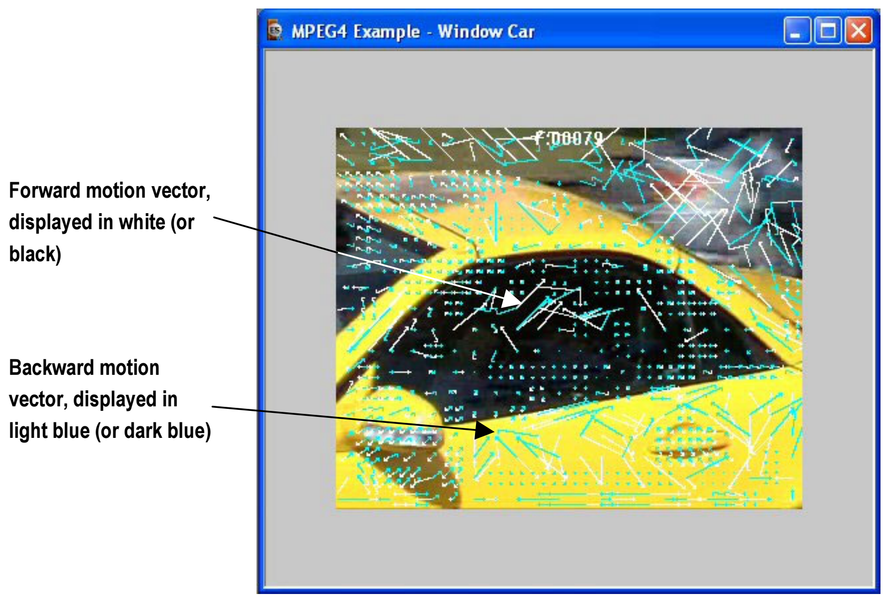
The motion vectors are plotted to the length of that actually used by the decoder.

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 bi-directionally 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:



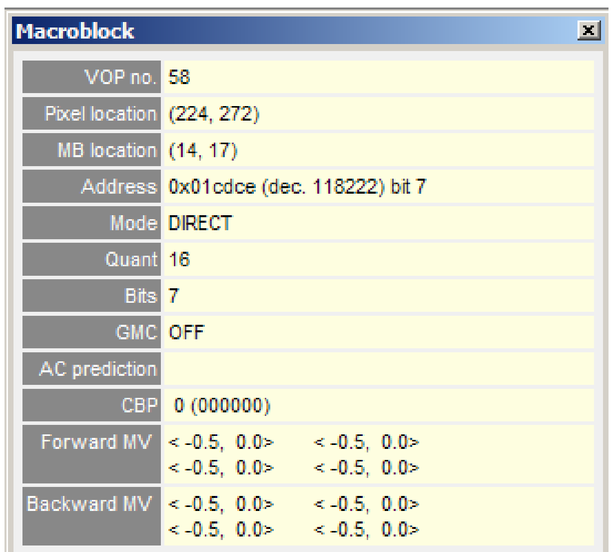
In Interlaced video sequences the motion vectors are displayed in the following colors:

MV Type	Field	MV color - out	MV color - out
Forward	Top	White	Black
Backward	Top	Light blue	Dark blue
Forward	Bottom	Yellow	Green
Backward	Bottom	Magenta	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 **Information** provided by MB Tooltip on page 6–79 (at the end of the 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:



MB statistics

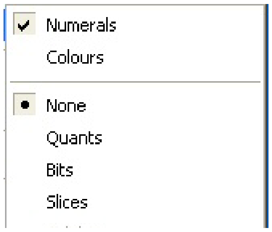
This option displays statistics on a MacroBlock-by-MacroBlock basis, either for the current frame only or averaged over a range of frames.

The top of this sub-menu allows selection of:

- Numerals (numbers)
- Colors

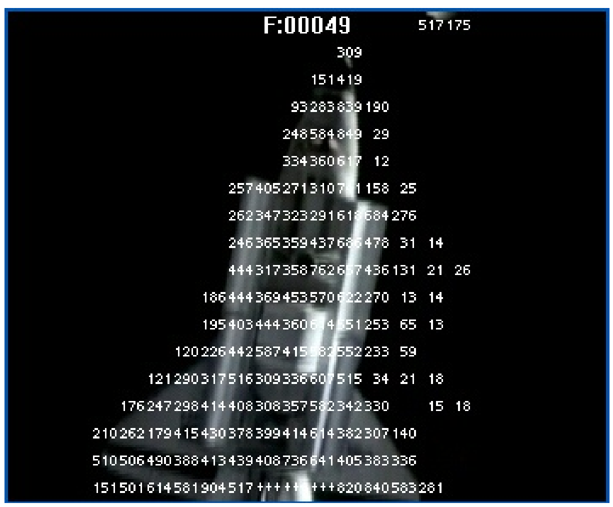
Numerals

When this is selected, numbers are displayed for the MacroBlock statistics, overlaid onto each MacroBlock.

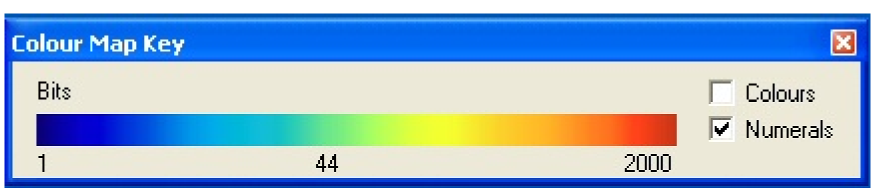


(In the above example, Numerals has been selected, but None is also selected, so no statistics will be displayed.)

The number displayed is selected from this sub-menu; the example below 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:



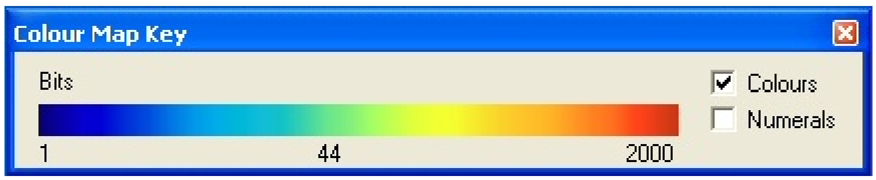
See the next section for description of the Color Map Key.

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:

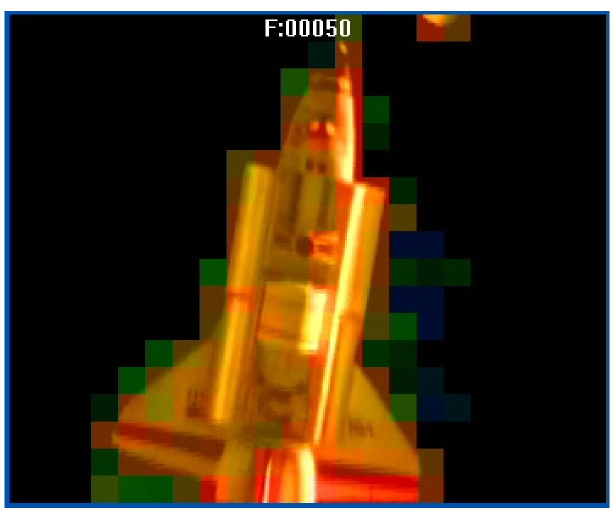


NOTE. If it is hard to see the colors, the video can be blanked (set to mid gray) by selecting Blank video in the Overlay menu.

If exact numbers are required, the Numerals can be switched on instead (or as well as the colors), or the MacroBlock tooltip can be used.

If it is not displayed, the Color Map Key can also be displayed from the Windows menu, Toolbars.

The example below shows bits per MacroBlock in frame 50 of the MPEG-4 example file Space (no color=no bits):



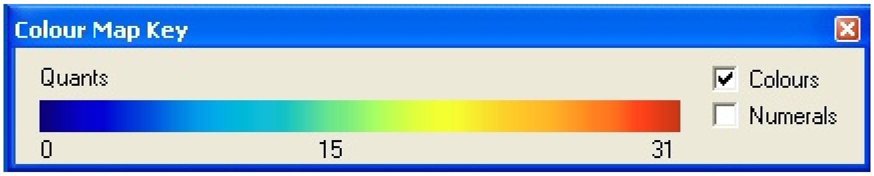
Linear/logarithmic scales

Some number ranges are best represented in linear scales (such as quants and slices); others 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 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, then a logarithmic scale has been chosen by MTS4EA).

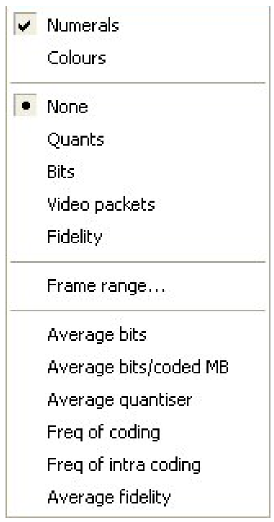
Linear scale (e.g. quants, slices):



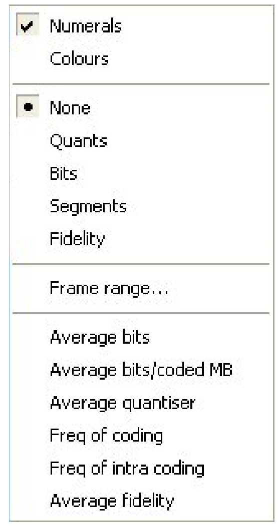
Logarithmic scale (e.g. bits, average bits):

Menu variation with standard

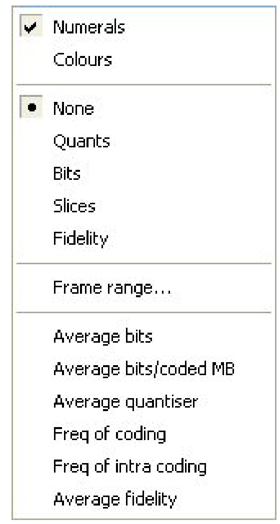
The menu below the selection of numerals/colors varies (slightly) with the video standard:



MPEG-4



H.263. H.261



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 6–214 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. *The 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 20-25, 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 a range of frames has been set (see **Frame** range on page 6–101).*

*The status bar indicates whether the video frame being displayed is inside or out of the selected range of frames (see **Frame** range in/out indicator on status bar on page 6–103).*

None. This option switches off any displayed overlaid MacroBlock statistics on the current frame.

NOTE. *The None option does not switch 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 a not coded MacroBlock.

NOTE. *As 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]

NOTE. *Video packets are only used in displays of information relating to 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.

Segments [H.263, H.261]

NOTE. *Segments are used only in displays of information relating to H.263.*

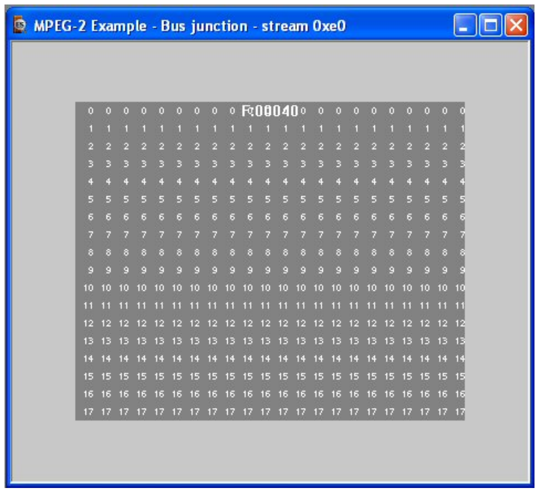
This option shows the segment numbering in use for the current frame. Segments are defined as being Group of Blocks (if GOB headers are sent in H.263) or Slices (as defined in Annex K of H.263+). If a segment number is zero then it is not displayed - hence if no GOB headers or Slices are sent, then this option will not overlay anything.

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]

***NOTE.** Slices are used only in displays of information relating to H.264/AVC and MPEG-2.*

This option shows the slice numbering in use for the current frame; an example is given below:



(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, e.g. PSNR in dB (decibels):

***NOTE.** Fidelity is grayed out until fidelity analysis is enabled; see* **Fidelity** *enable* on page 6–139.

The fidelity analysis done (which type of PSNR, etc. analysis) is given in the Fidelity tab of the Analysis options - see **Fidelity** enable on page 6–139 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):

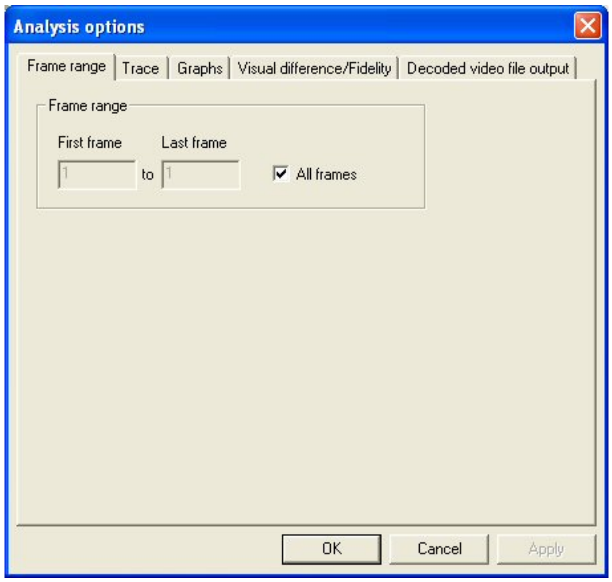
- Average bits
- Average bits/coded MB
- Average quantizer
- Freq of coding
- Freq of intra coding
- Average fidelity

***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 6–130).*

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.

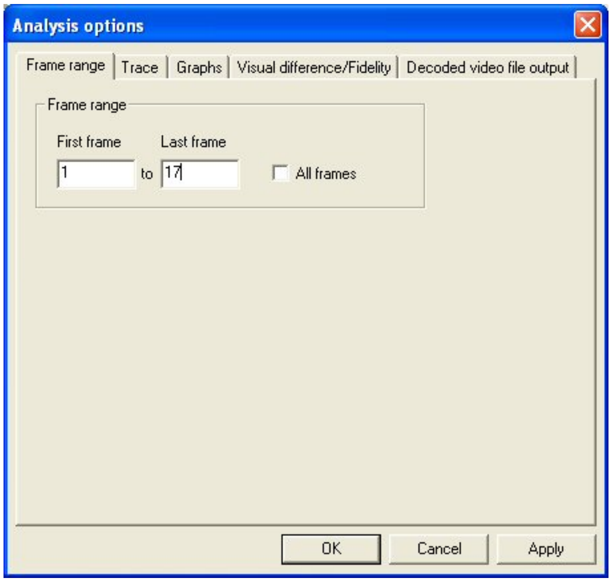
NOTE. *The frame range over which statistics are to be accumulated cannot be changed during pause mode; otherwise MTS4EA could not ensure that the Trace and Graph data were collected over the correct range of frames. For example, if the video was paused at frame 23 and Trace was enabled for a range of frames from 20-25, the Trace data would be incorrect.*

Clicking on Frame range takes the user to the frame range tab:

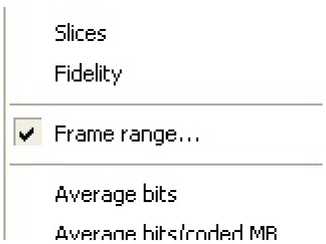


All frames. This option is the default and calculates the sequence statistics from the start of the sequence until the current frame.

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):



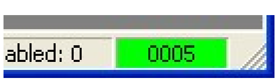
When a range of frames has been set, the Frame range... item on the MB statistics menu changes, with a check mark to indicate a range of frames has been set:



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 below:



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. *As 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).*

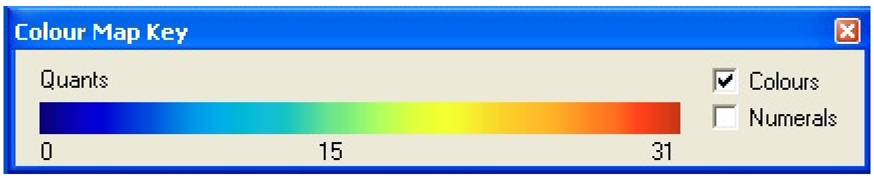
The color range displayed with the color overlays is a logarithmic scale:

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 above.

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:



Freq of coding. This option displays the frequency of coding (number of times each MacroBlock has been coded over the frame range specified).

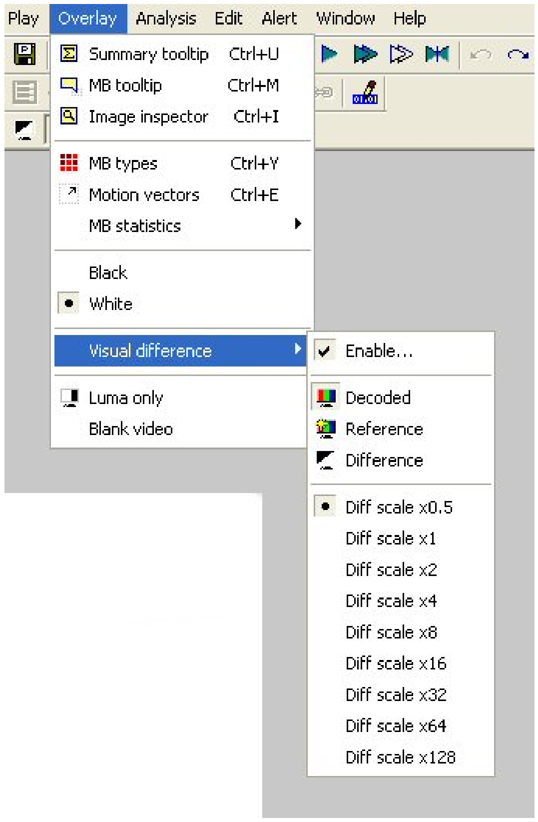
NOTE. *As there is only enough space to display three digits, if the MacroBlock used more than 999 bits, then +++ is displayed to signify an overflow (a number greater than 999).*

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 fidelity analysis done (which type of PSNR, etc. analysis) is given in the Fidelity tab of the Analysis options - see **Fidelity** enable on page 6–139 for more information.

Visual difference



This menu selection 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 can be done on a range of frames only, provided this range is at the start. 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 forwards/backwards (although some of the seek functions, e.g. to seek to an I-frame, are not relevant to uncompressed video files and therefore will be grayed out).

Visual difference icon toolbar

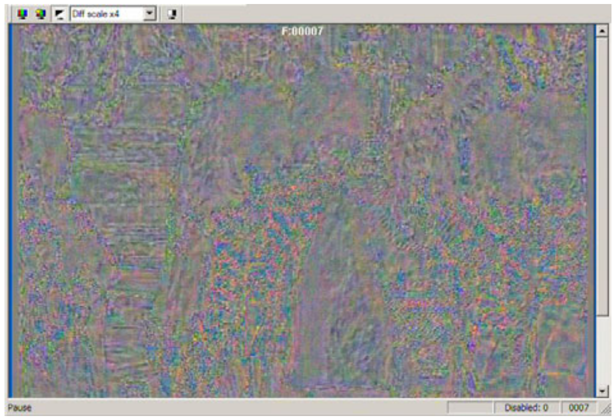
This toolbar is displayed when the visual difference view is enabled:

The functions of the icons are shown below.

Show encoded (compressed). Selecting this menu option (or clicking on the  icon) displays the standard video window (shows the compressed bitstream which has been decoded by 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. This subtracts the uncompressed video reference video from the compressed video, frame-by-frame:



MTS4EA uses the timing given in the:

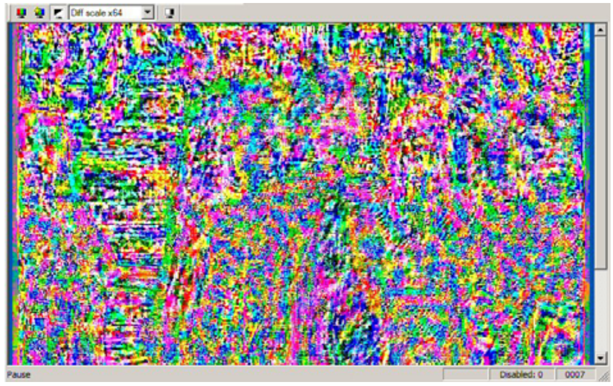
- Compressed bitstream itself for the displayed frame times of the compressed bitstream
- Uncompressed video reference set-up tab of MTS4EA (the frame rate) for the uncompressed video reference file

When doing the subtraction, 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:

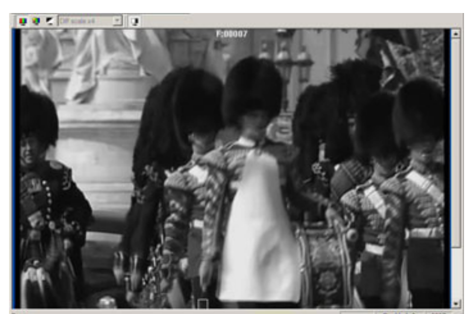
Diff scale x0.5
Diff scale x1
Diff scale x2
Diff scale x4
Diff scale x8
Diff scale x16
Diff scale x32
Diff scale x64
Diff scale x128

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

Black/White `Ctrl+W`

This item allows the selection of the color of the overlays. It allows the user to 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

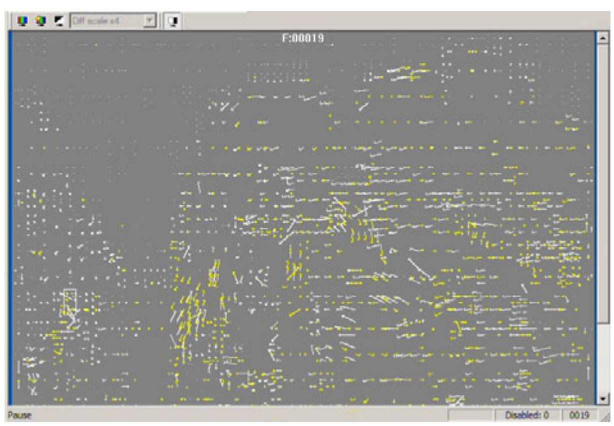
See **Motion** vectors `Ctrl+E` on page 6–91 for more information.

Black. This selects a black/dark blue color for the overlays.

White. This selects a white/light blue color for the overlays.

Blank Video

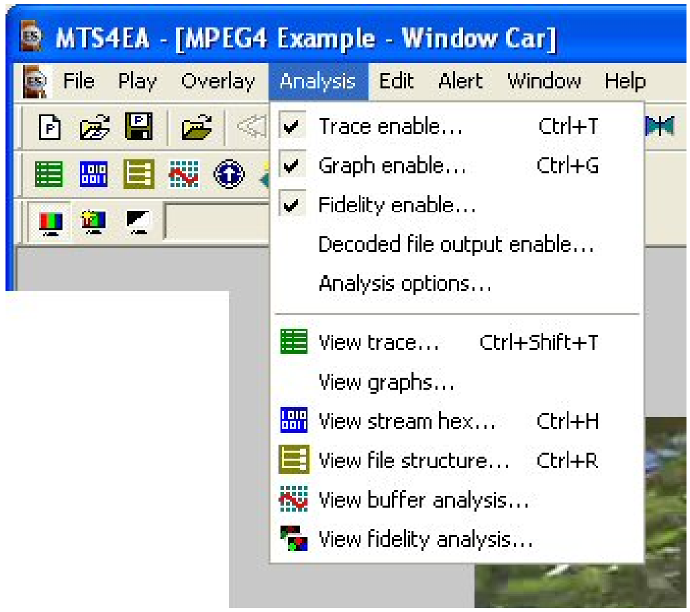
This option will blank the video output to a gray color. It is included so that the motion vectors and other data can be viewed more clearly without the visual interference of the decoded image.



In this example, the motion vector overlay is switched on and the video is blanked.

NOTE. *The Blank video option is available only when one of the overlays on the Overlay menu is being use; otherwise it is grayed out.*

Analysis Menu



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:

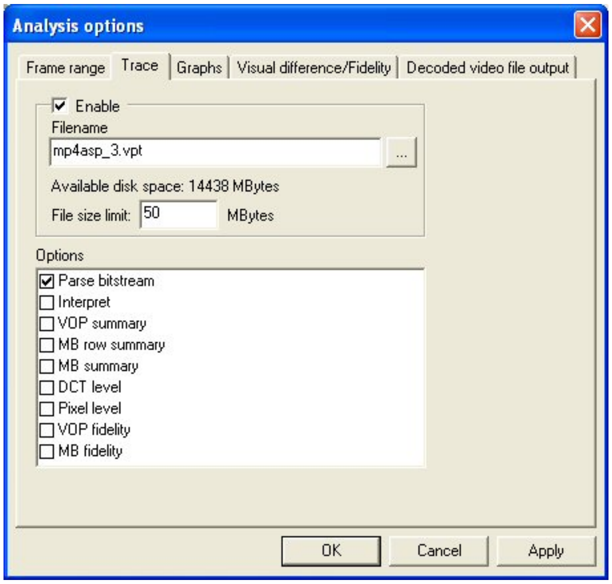
- Hex (hexadecimal) view of stream data
- The structure of the file being analyzed
- Analysis of the video buffer usage

This menu is enabled for video streams only.

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.

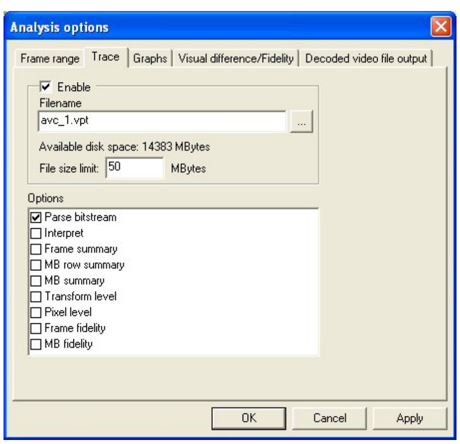
NOTE. *Trace enable..., Graph enable..., Fidelity enable..., Decoded file output enable... and Analysis options... cannot be set in pause mode - the video must be stopped, otherwise MTS4EA could not ensure that the Trace, Graph and Fidelity data were collected over the correct range of frames.*

Trace enable `Ctrl+T`

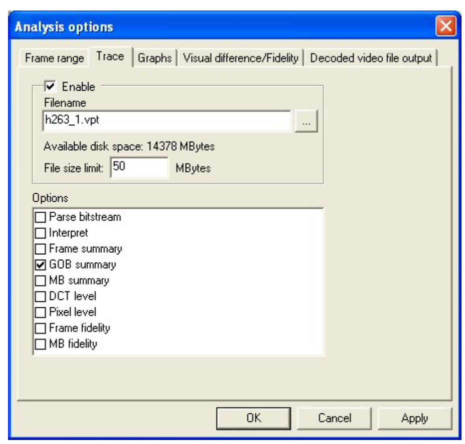


The Trace options provide text outputs that describe the contents of an encoded video stream, to various levels of detail.

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.



H.264/AVC



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, the user 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 6–116 below.

***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. See below.

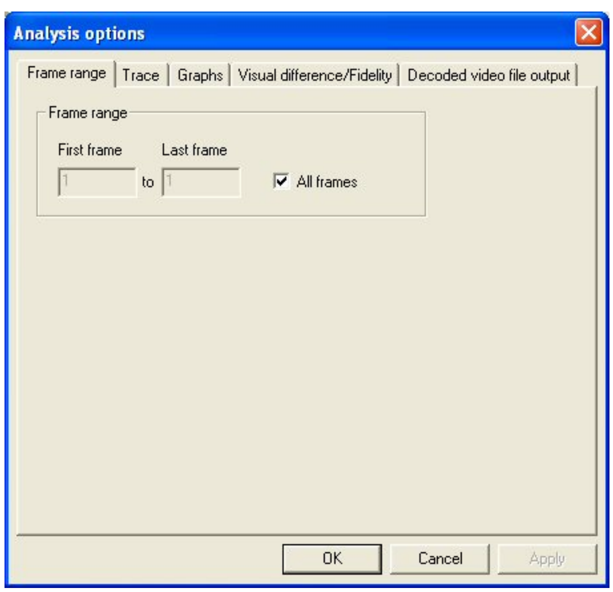
Frame range tab

This option allows the user 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 6–101).*

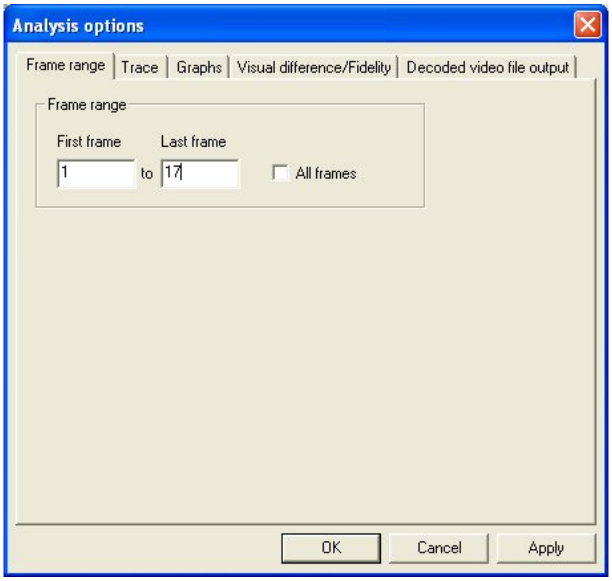
The frame range over which statistics are to be accumulated cannot be changed during pause mode; otherwise MTS4EA could not ensure that the Trace and Graph data were collected over the correct range of frames. For example, if the video was paused at frame 23 and Trace was enabled for a range of frames from 20-25, the Trace data would be incorrect.

Clicking on Frame range takes the user to the frame range tab:

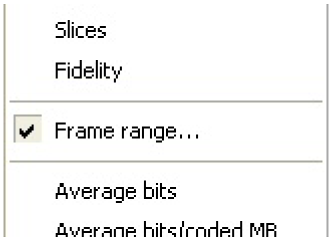


All frames. This option is the default and calculates the sequence statistics from the start of the sequence until the current frame.

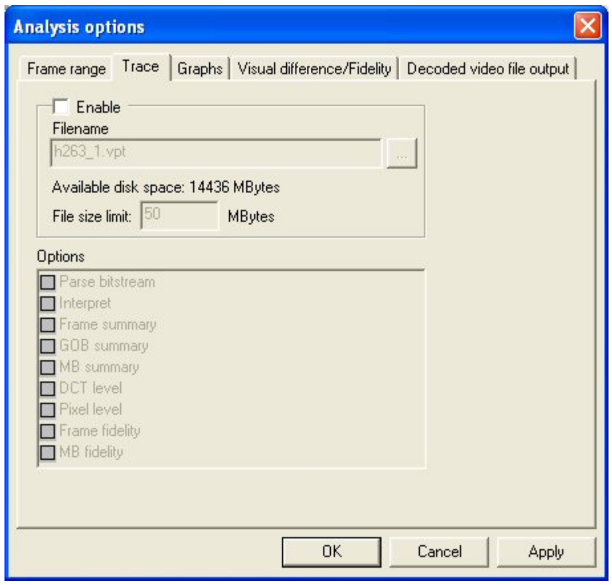
To set a range of frames. If the All frames check box is cleared, then a range of frames can be entered (in this case, frames 1 to 17 inclusive):



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:



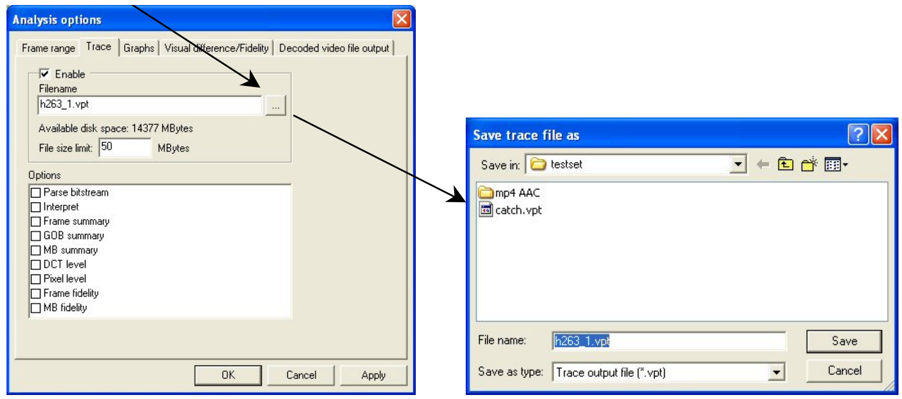
Enable. This option Enables/Disables the Trace output. If the output is disabled, the remaining selections on the Trace tab are unavailable (grayed out) although the values are stored during the current use of MTS4EA. (They are reset to their default values the next time MTS4EA is started.)



Filename. By default, the filename suggested is the same as the base name of the input video file (the video filename without the .m4v, etc. extension).

Any other filename can be entered; 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:



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 the user to limit the amount of disk space taken by the Trace file.

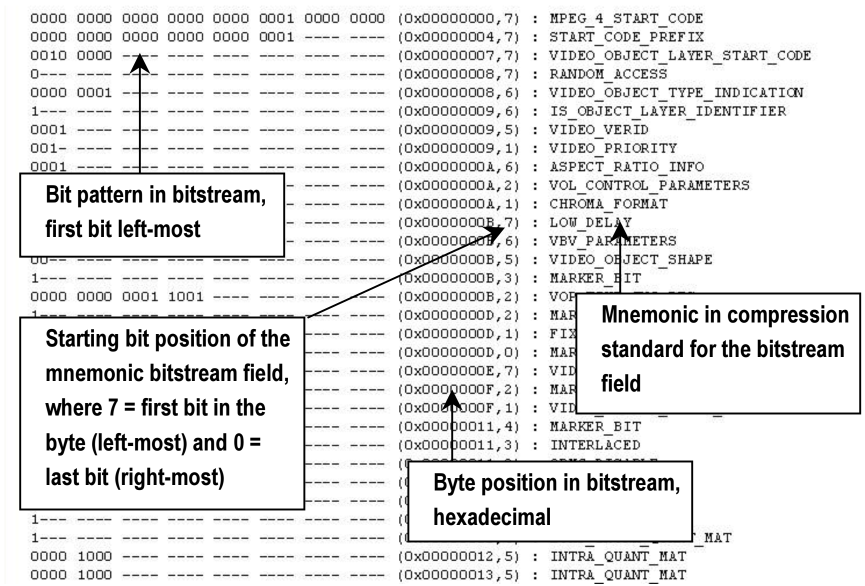
Once the Trace file reaches this size, it stops writing more data.

Parse Bitstream

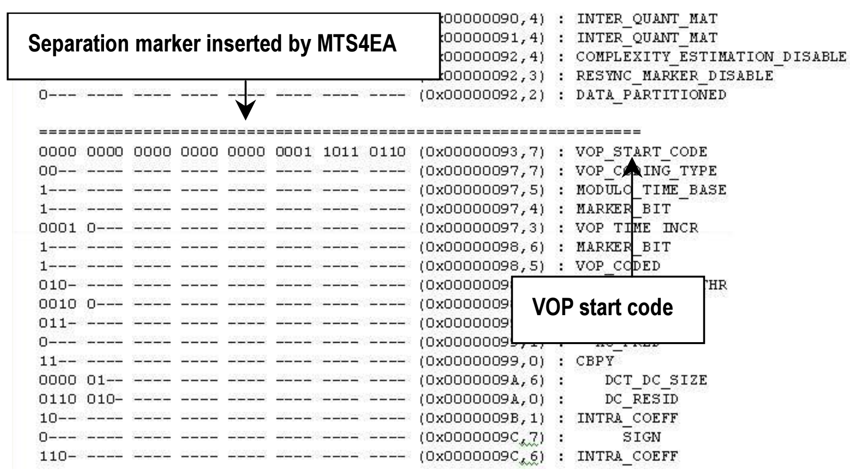
If this option is selected, then the bitstream is parsed to a file. This gives the individual bit patterns for the bitstream fields as well as the mnemonic used within the standard to identify the field.

NOTE. *The data from the output of Parse Bitstream is one of the best ways that MTS4EA provides of doing bitstream syntax debugging.*

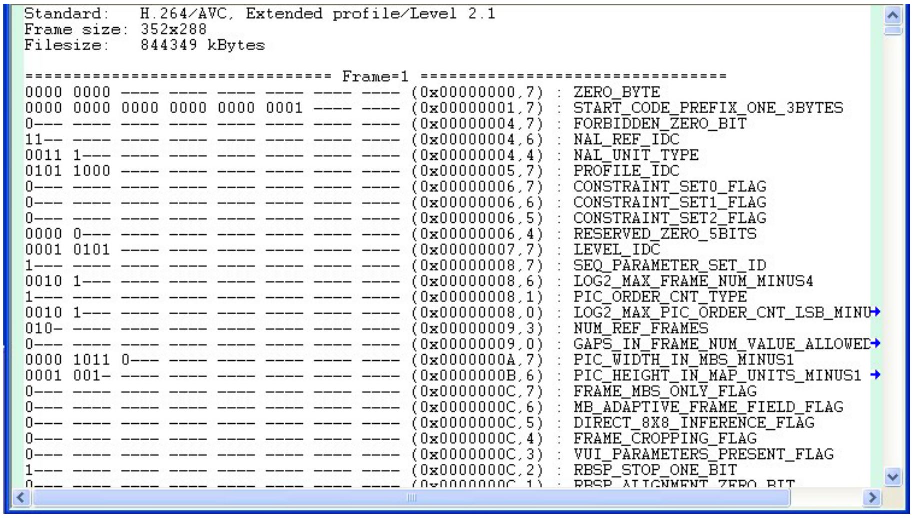
*See **General** codes used in Trace files and Alerts on page 7–1 for detailed information on the meaning of these.*



MPEG-4 example, at start of bitstream

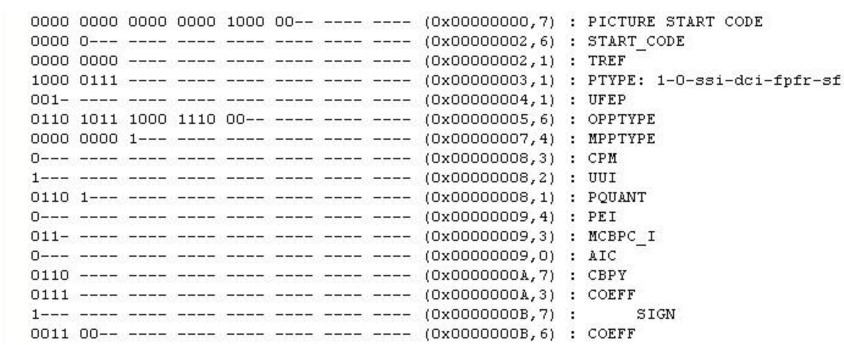


MPEG-4 example, at start of VOP



```
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
```

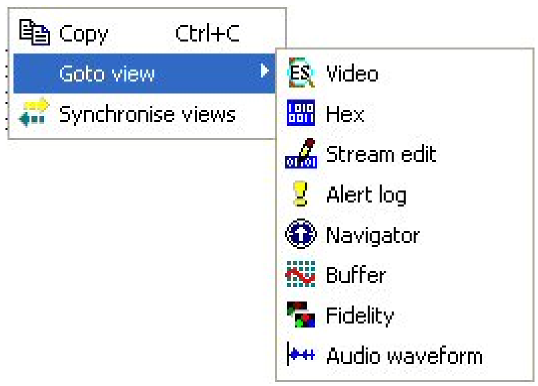
H.263 example, at start of bitstream

NOTE. *Also check out Interpret below as another excellent tool to do bitstream syntax debugging.*

To quickly find 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:

- 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 which has been selected in the Trace file:

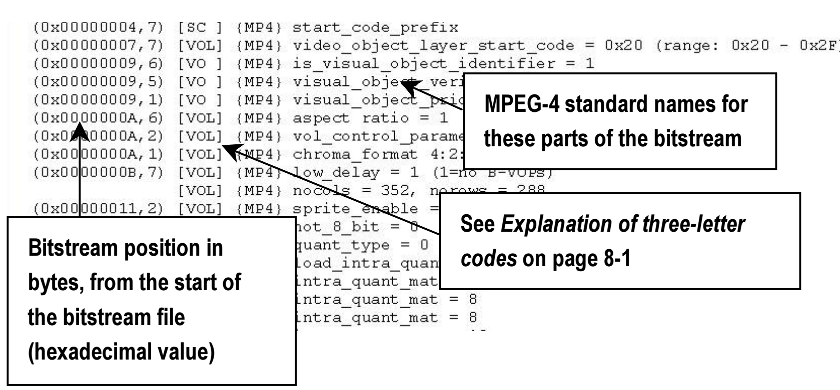


NOTE. This right-click context-sensitive menu is only active in the Trace/ Interpret and Trace/Parse Bitstream trace files.

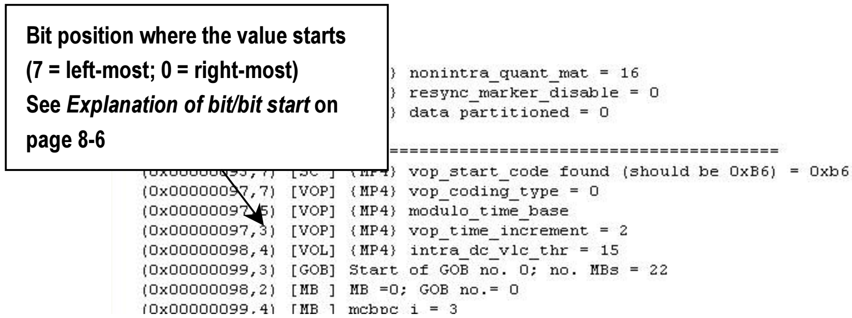
Interpret

This provides interpretation of the decoded data in the bitstream. Example outputs are shown over for MPEG-4 and H.263:

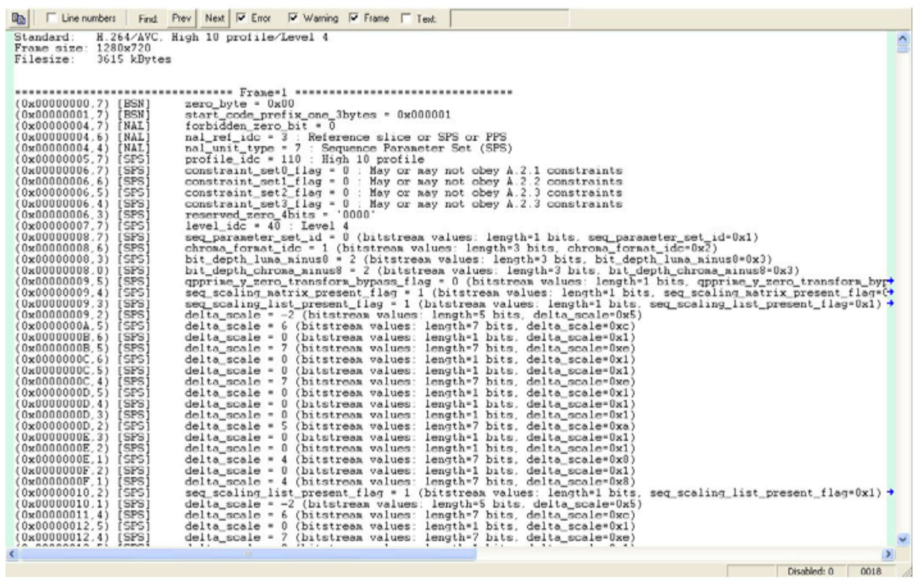
***NOTE.** It is important to also see **General** codes used in Trace files and Alerts on page 7–1 for an explanation of some of these items.*



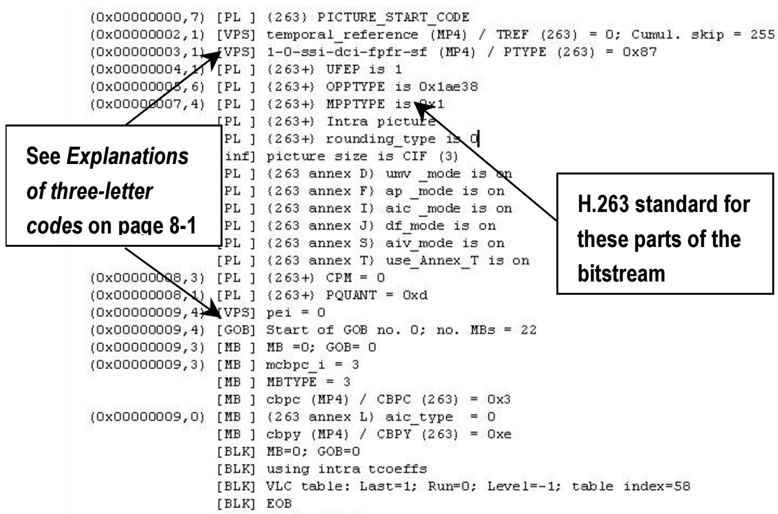
MPEG-4 example, at start of bitstream



MPEG-4 example, at start of VOP



H.264/AVC example, at start of byte stream (full MTS4EA Trace view window)

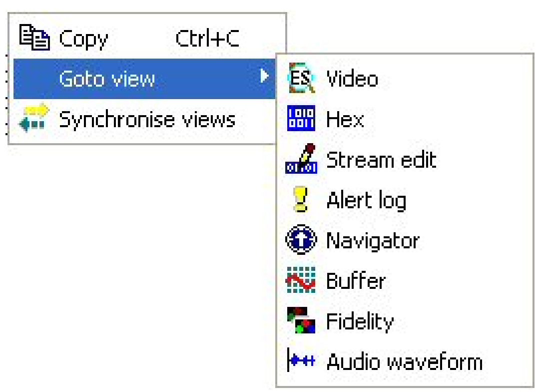


H.263 Example at start of bitstream

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.*

By right-clicking with the mouse, 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
- Go to the area in another view which corresponds to the area which has been selected in the Trace file:

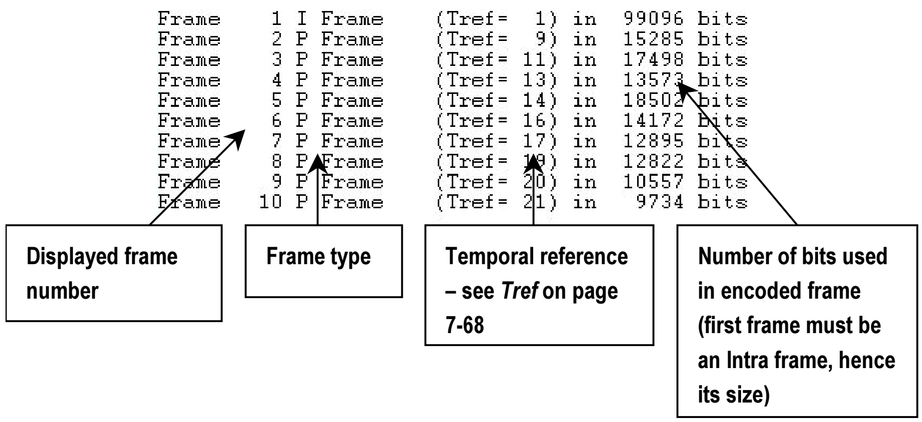


NOTE. *This right-click context-sensitive menu is only active in the Trace/ Interpret and Trace/Parse Bitstream trace files.*

Frame Summary [H.264/AVC, MPEG-2, H.263 and H.261] 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.

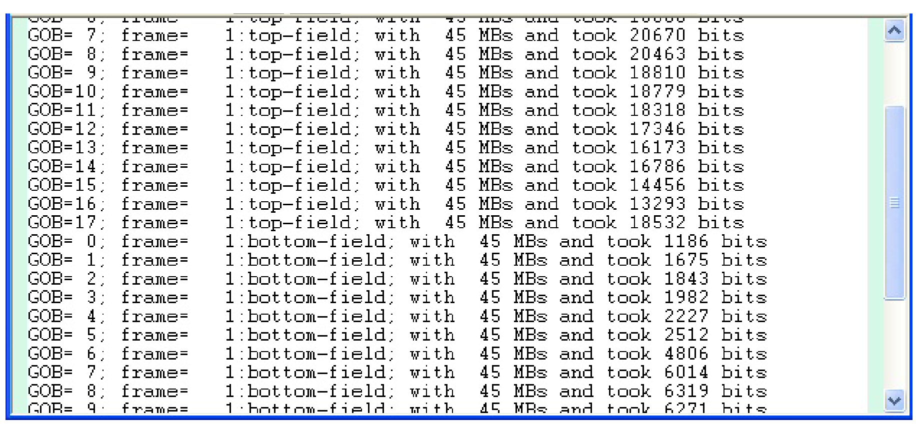
This example is for H.263, where Tref is used.



MB row summary [H.264/AVC, MPEG-4, MPEG-2] or **GOB Summary** [H.263, H.261]

This Trace outputs summary data for each horizontal row of MacroBlocks, starting from the top of the frame and working downwards.

NOTE. *A GOB in H.261 may not be a single horizontal row - for H.261, a GOB can start anywhere in one row and finish anywhere in the same row or subsequent rows.*



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, H.261 only]* on page 6–81)

MB Summary

This option outputs for each MacroBlock 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, H.263 and H.261]

NOTE. *In H.264/AVC the report is of transform coefficients that are not DCTs.*

For MPEG-4, H263 and H.261 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
```

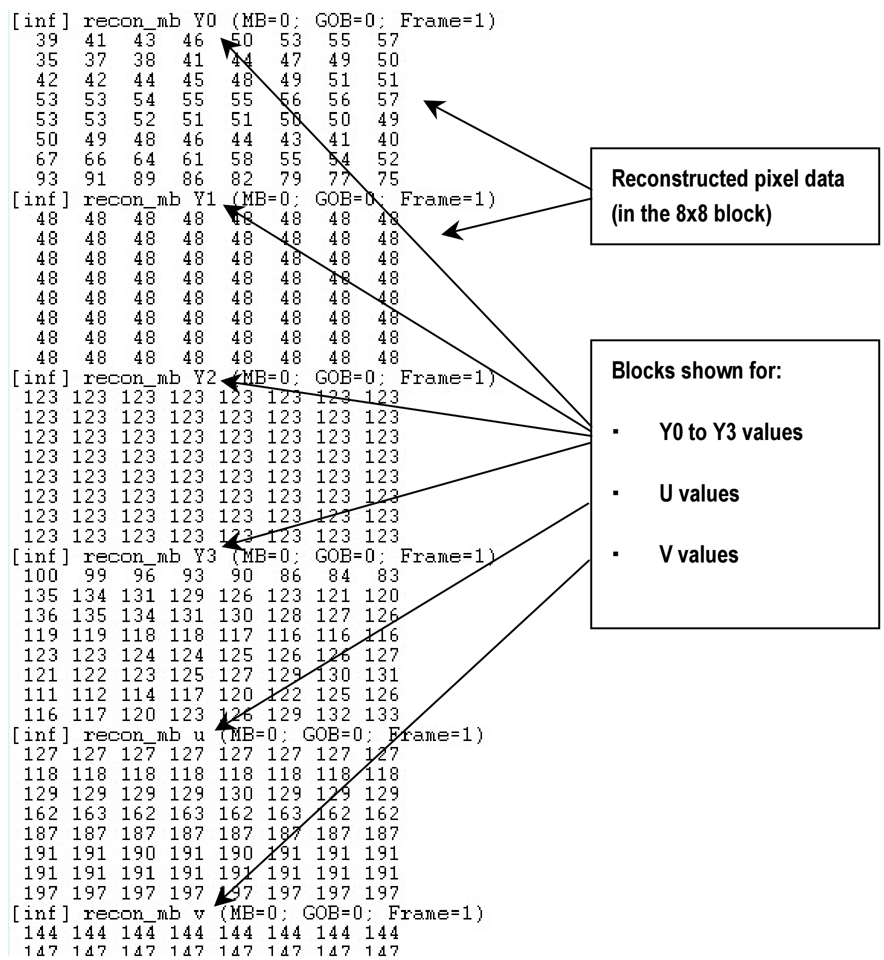
In MPEG-2 the data is reported in the following format:

```
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.



***NOTE.** This can also generate a lot of data.*

Frame fidelity [H.264/AVC, MPEG-2, H.263 and H.261] or **VOP fidelity** [MPEG-4]

This provides a Trace output of the fidelity analysis frame-by-frame. See **Fidelity** enable on page 6–139 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 screenshot above for frames 10 and above are empty as 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 6–40.*

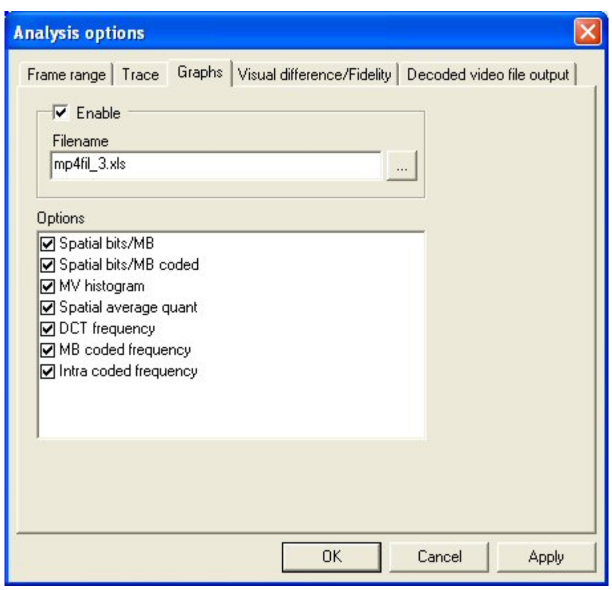
MacroBlock fidelity

This provides a Trace output of the fidelity analysis for each MacroBlock in the frame range. See **Fidelity** enable on page 6–139 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 ***YUV*** *source files for example fidelity analysis on page 6–40.*

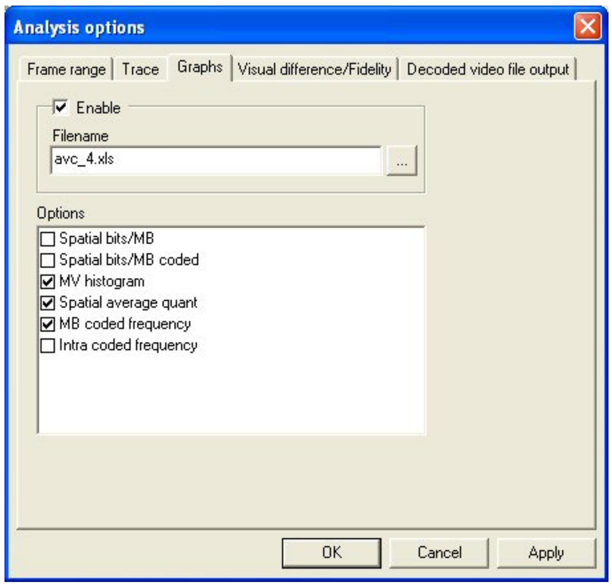
Graph enable `Ctrl+G`



Graph enable... switches 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 is called which converts the statistics data into graphs, which are then available in various tabs of Excel.

Note that the collection of statistics varies somewhat depending upon the standard concerned - for example, there is no DCT as such in H.264/AVC, so this is not available (as shown below):



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. *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.*

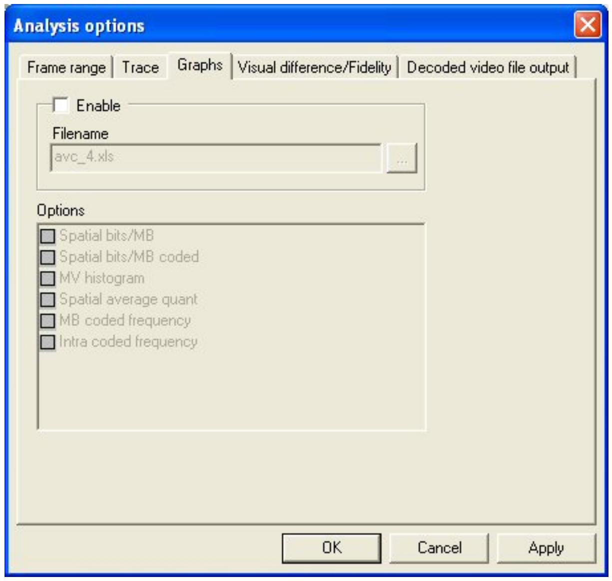
Frame range tab

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 6–113.

Enable

This option enables/disables the collection of the statistics for the Graph data. When disabled, the Graph tab appears as below:



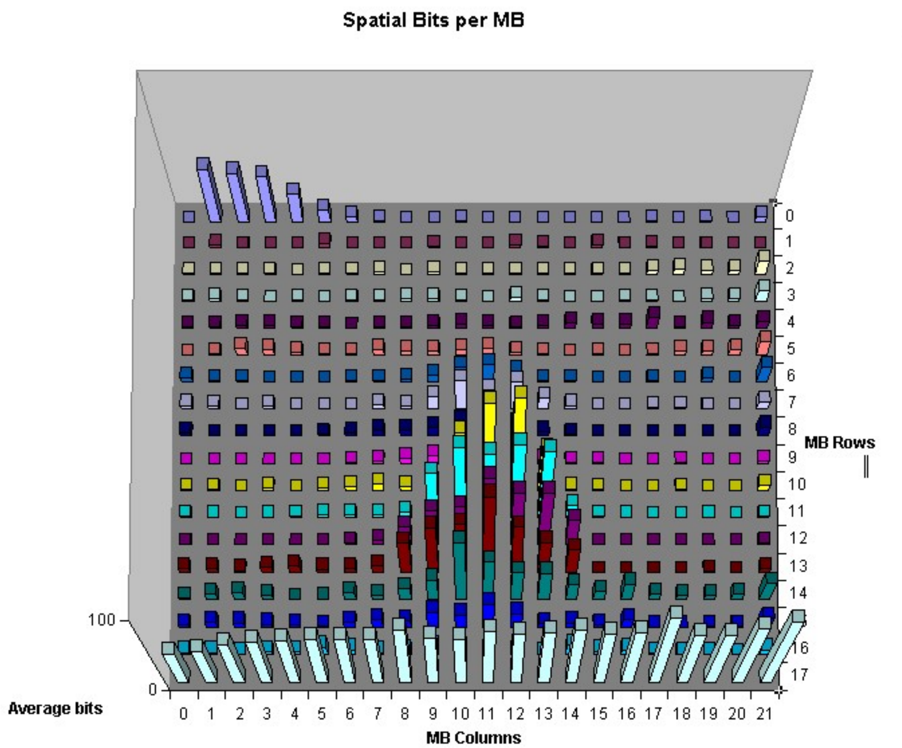
Filename. By default, the filename suggested is the same as the base name of the input video file given the extension `.xls`.

A different filename can be entered (but not a different file extension), and the folder where the file is to be stored 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.*

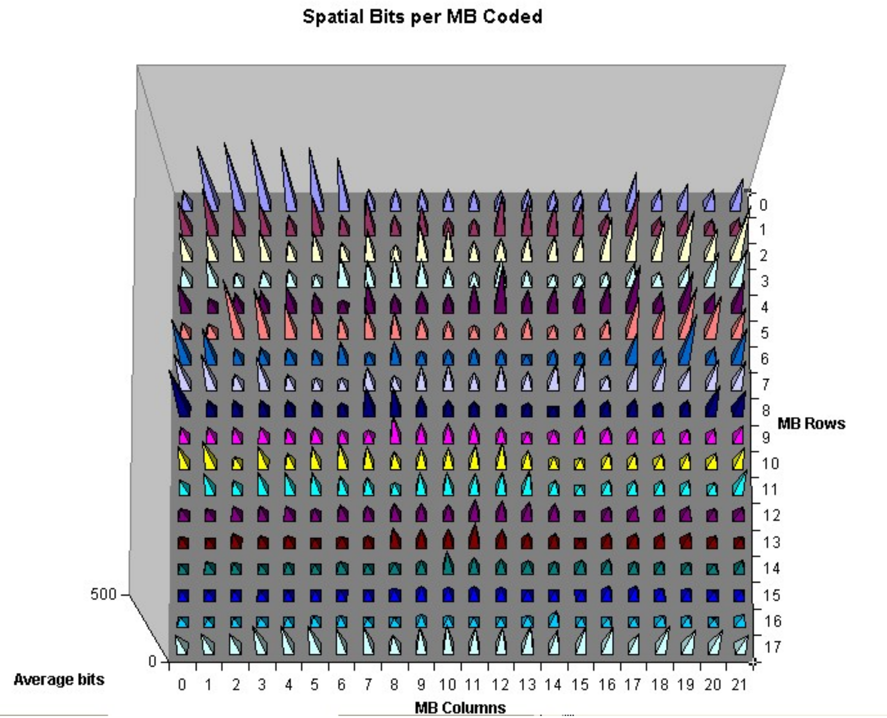
Spatial bits/MB

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.



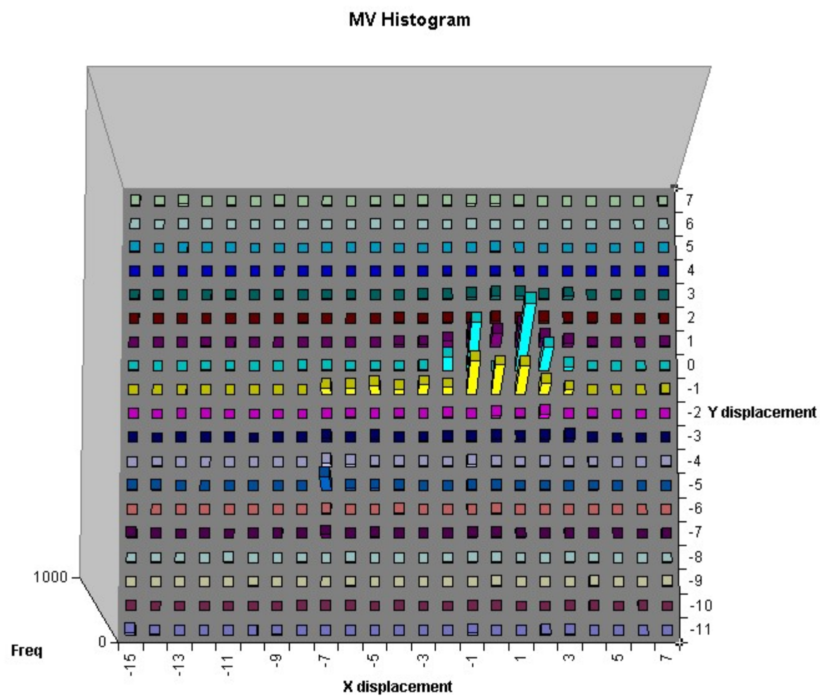
Spatial bits/MB coded

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.



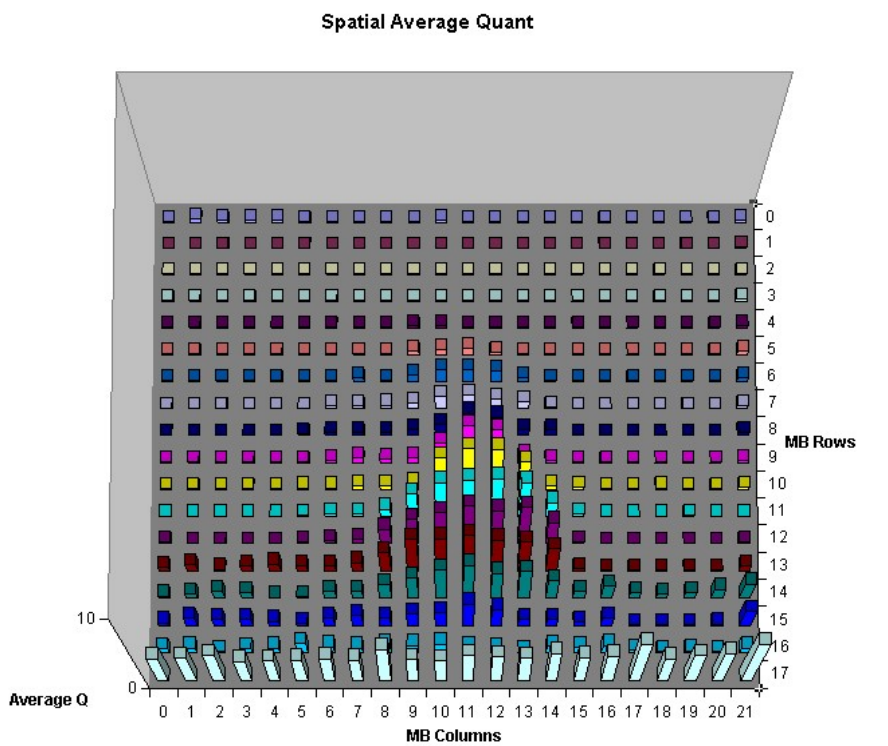
MV histogram

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 have been chosen.



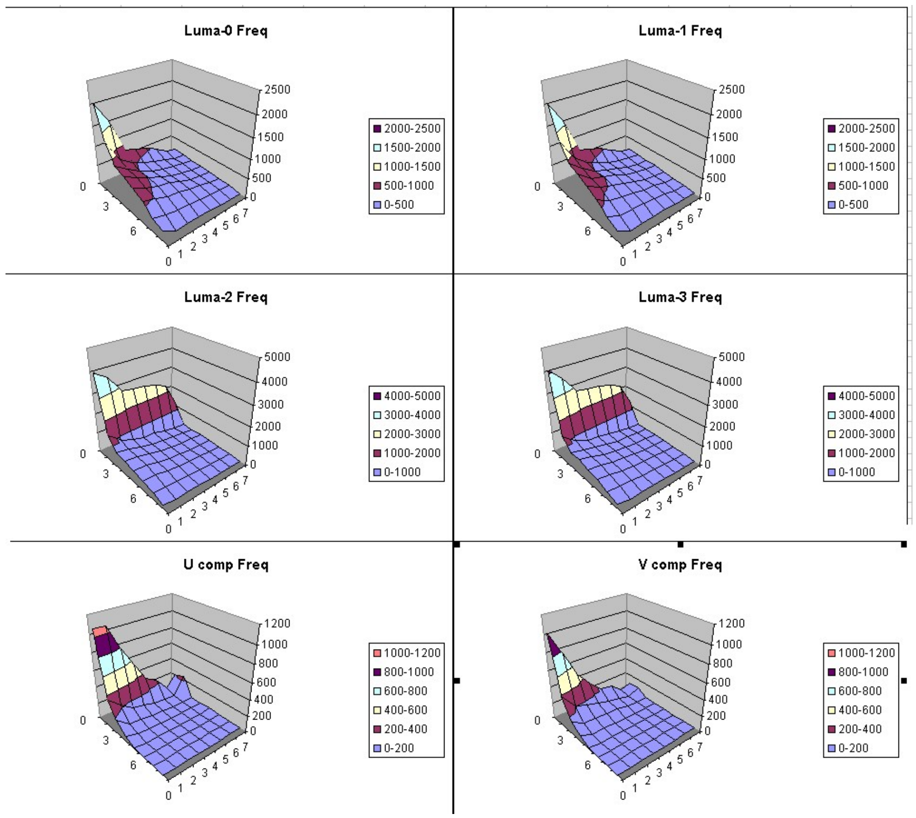
Spatial average quant

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).



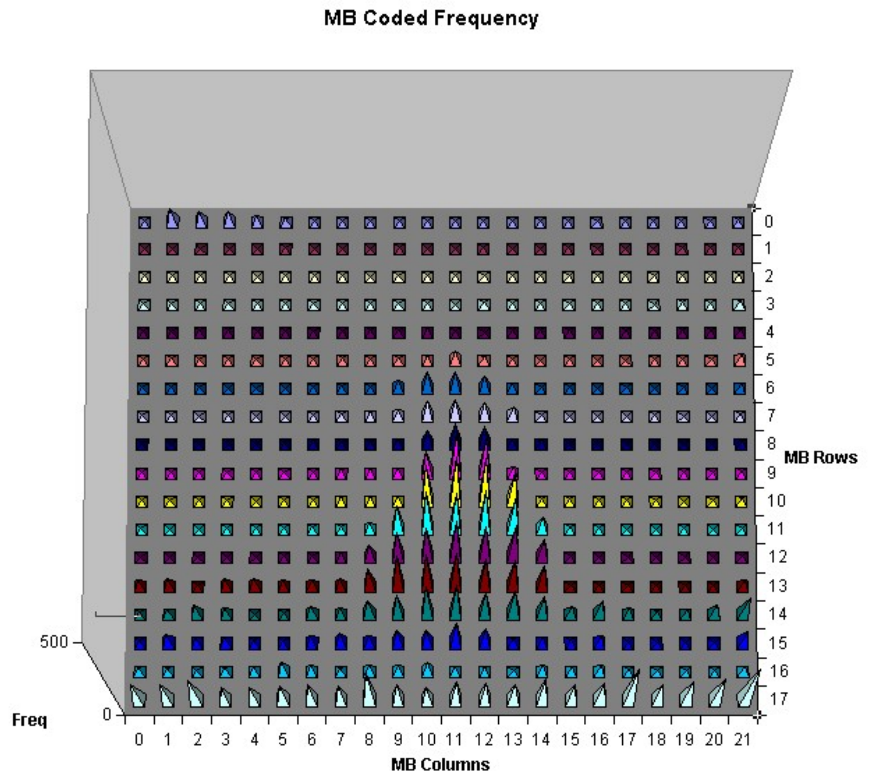
DCT frequency [MPEG-4, MPEG-2, H.263 and H.261 only]

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 comprise 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.



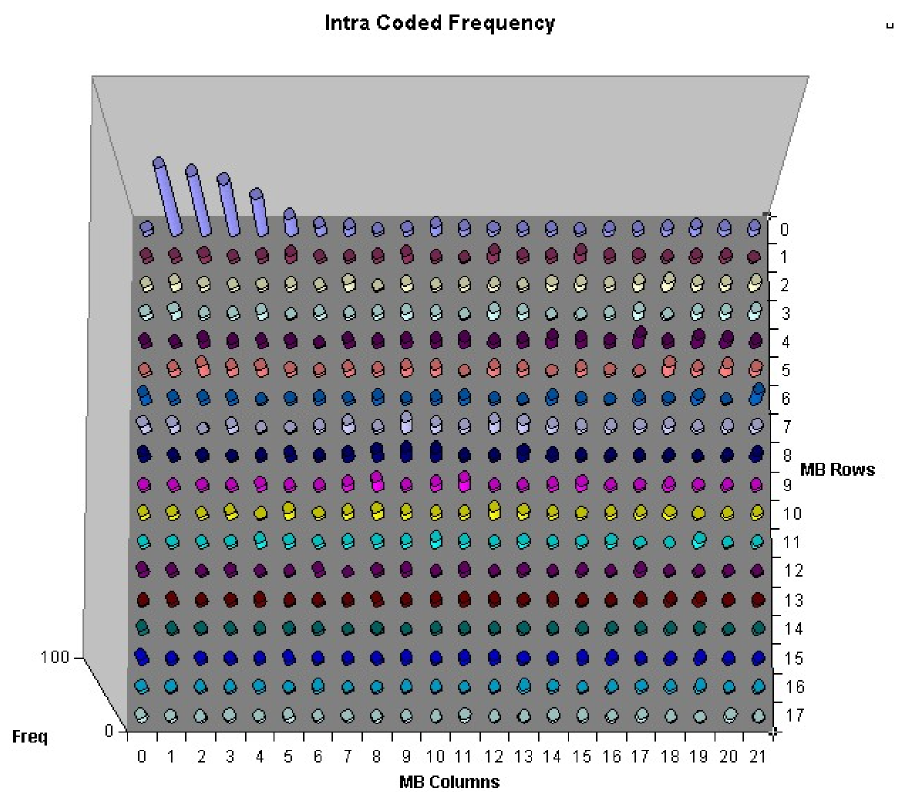
MB coded frequency

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 either issues in the digitization process, the rate control process or the sensor.



Intra coded frequency

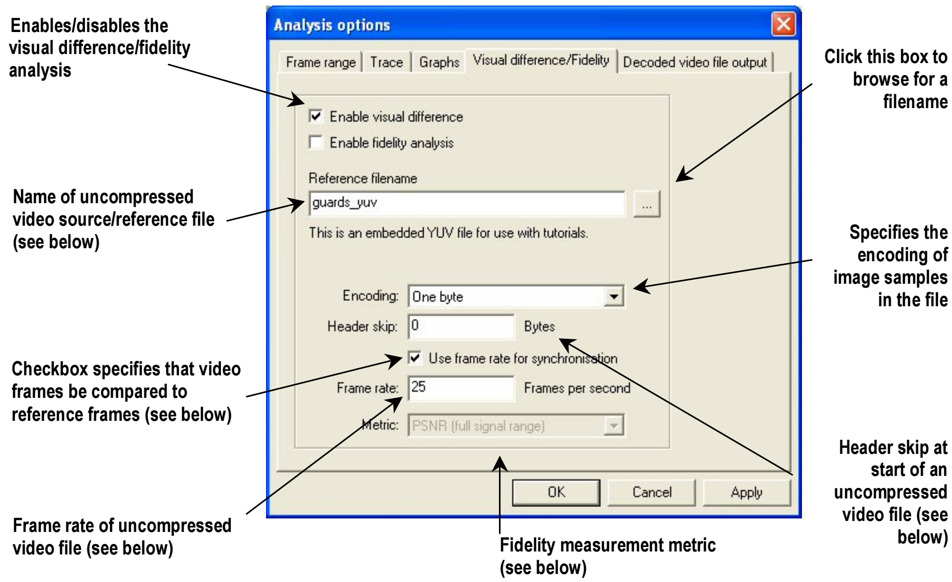
This option creates a graph of the frequency of intra coding of each MacroBlock. Over time, each MacroBlock may be updated in Intra mode every so often (as required in the Standards) in case there is a line error, or a mismatch between encoder and decoder. Encoders which do this more often are more likely to have better error robustness, although at the expense of more bits being used (as intra coding takes more bits than inter coding).



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:



The results of the fidelity analysis are displayed:

- In the Trace files, when the appropriate Trace option is selected (see **Trace** enable `Ctrl+T` on page 6–112, *Frame fidelity* on page 6–128, and **MacroBlock** fidelity on page 6–129)
- As real-time overlays, when the overlay is selected (see **MB** statistics on page 6–93, *Fidelity* on page 6–101, and *Average fidelity* on page 6–105)

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] under **Standards** References on page 5–20 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 6–34 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 prior to 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.

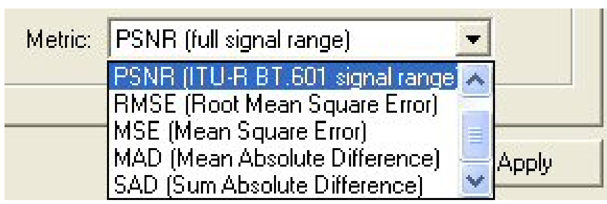
Frame rate. The rate at which the uncompressed video frames were recorded, in frames per second.

__NOTE.__ The number entered in the Frame rate field can be an integer (e.g. 30), or a fraction (e.g. 30000/1001) or a decimal number (e.g. 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:



These are all explained below.

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 equations given below.

For the explanations of fidelity metrics, the following nomenclature is used:

$\lvert \ldots \rvert$	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.

PSNR = 10 · log (S2/ 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:

PSNR255 = 20 · log (255/ 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:

$PSNR_Y = 20 \cdot \log (220/ RMS)$

$PSNR_{U,V} = 20 \cdot \log (225/ RMS)$

RMSE (Root Mean Square Error)

The RMS is the defined as the square root of MSE.

RMS = √ 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.

$MSE = 1/N \, \Sigma \, [f(x, y) - g(x, y)]^2$

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.

$MAD = 1/N \, \Sigma \, | f(x, y) - g(x, y) |$

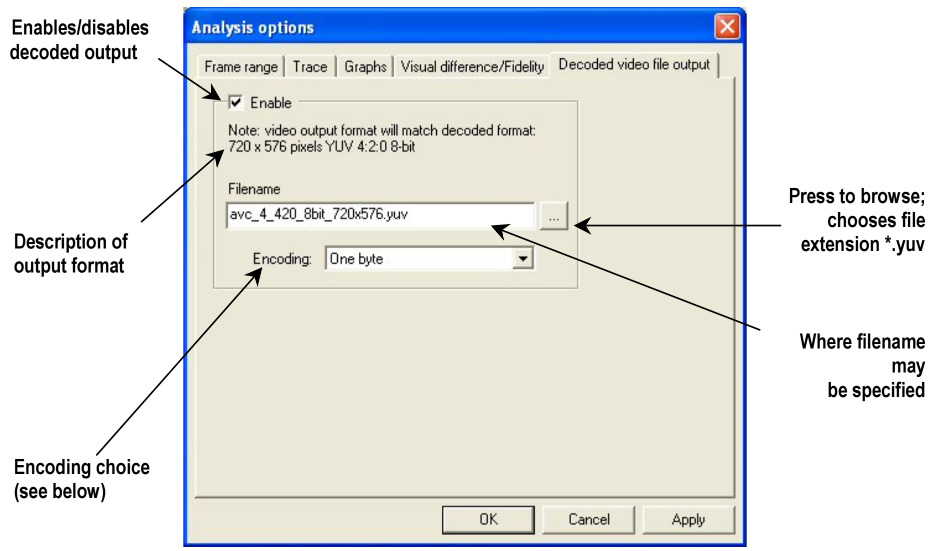
SAD (Sum Absolute Difference)

The SAD metric is defined as the sum of absolute differences between samples in the reference and decoded image planes.

$SAD = \Sigma \, | f(x, y) - g(x, y) |$

Decoded file output enable...

This opens the Decoded output tab of the Analysis options and allows the user to write a decoded file to disk.



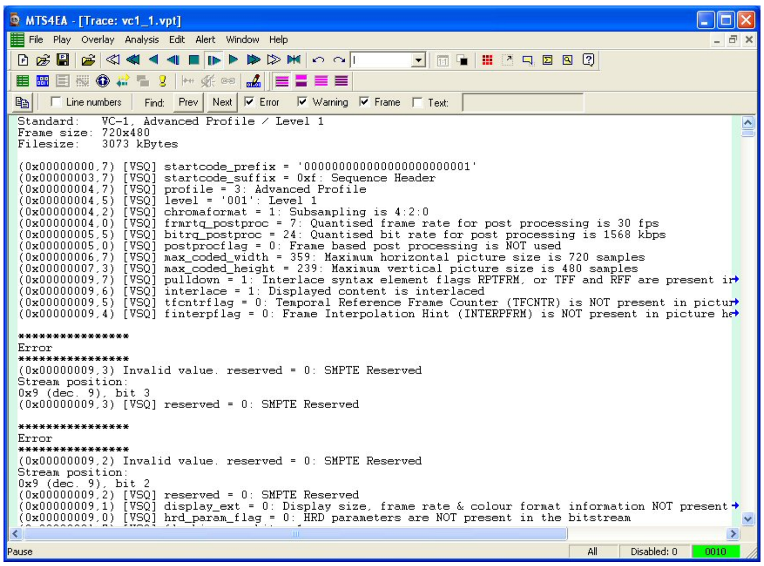
Encoding. This control specifies the encoding of image samples in the file format. The following options are available:

- 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

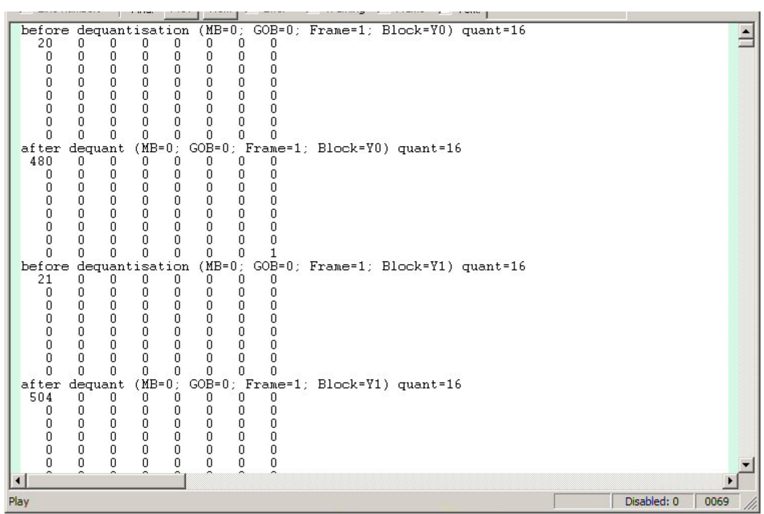
View trace... `Ctrl+Shift+T`

This option opens the Trace view on the current output Trace file, as given 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:



This is a Trace/DCT level example:



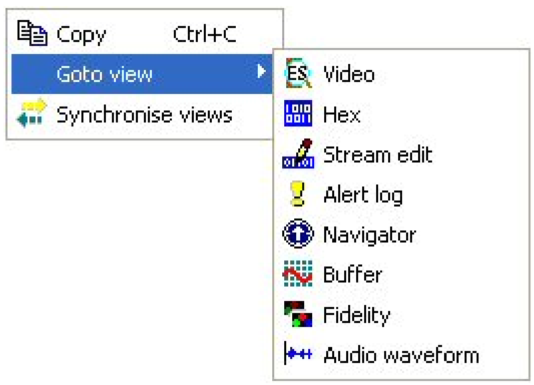
NOTE. *If there is not a current Trace file (and Trace enable is not enabled) then this menu option/toolbar icon is unavailable, grayed out.*

NOTE. For users of earlier versions of MTS4EA, this Trace viewer replaces the previous Trace view done using Microsoft Word.

If the video stream has been re-run 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

By right-clicking with the mouse, a context-sensitive menu appears which 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.

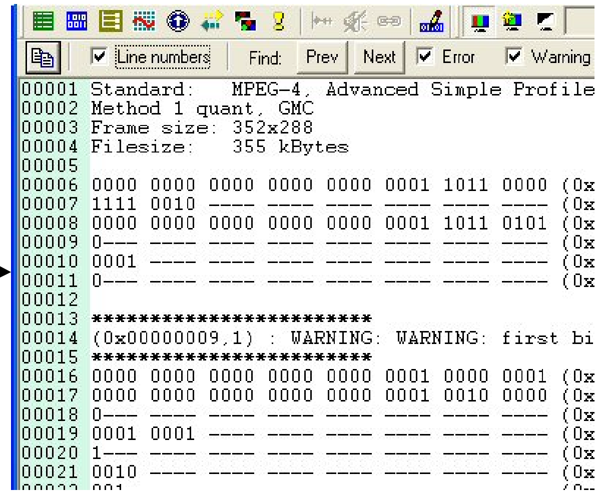
This can be done in a number of ways:

- By pressing `Ctrl+C`
- By right-clicking with the mouse and selecting Copy
- By clicking the copy icon (see below)

Line numbers

Line numbers can be switched on/off:

Line numbers displayed



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.

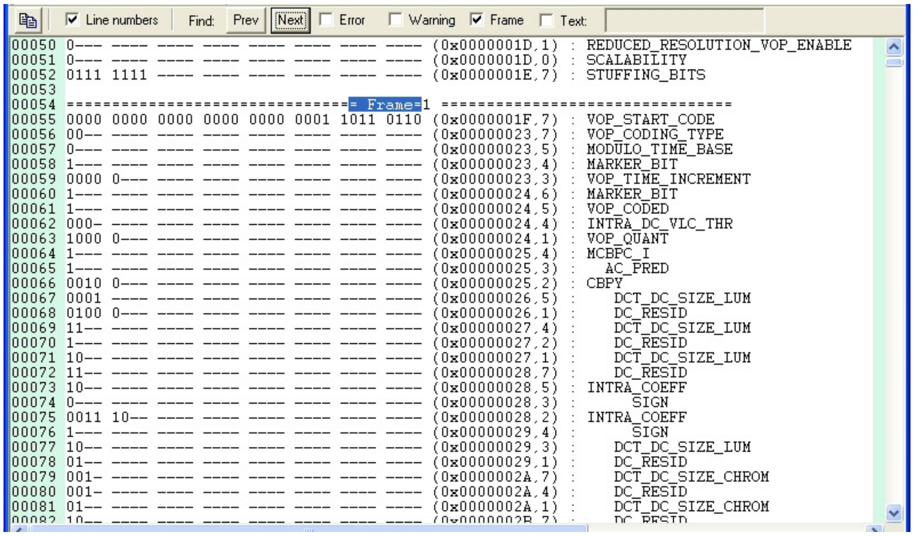
***NOTE.** The F3 key can also be pressed, to find next.*

The Shift+F3 key can also be pressed, to find previous.

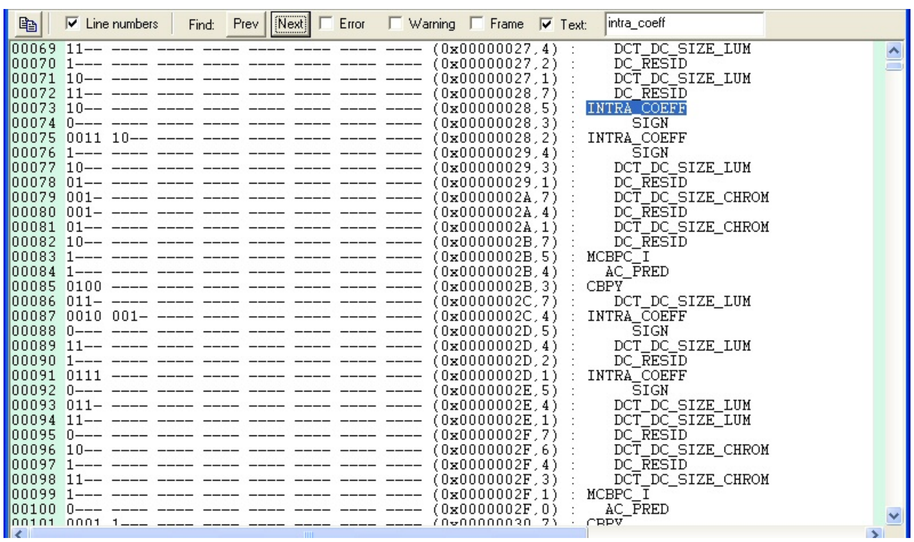
There are some standard strings that are useful to search for:

- Error
- Warning
- Frame

These are found by enabling the relevant check box (Frame example below):

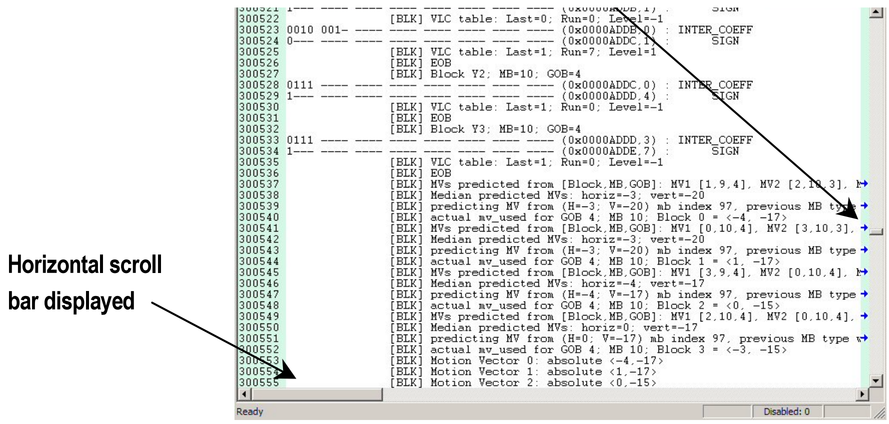


To search for any text, select the box next to Text and enter the text in the box (in this example, searching for `intra_coeff`):



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 the line extends to the right:



Trace file format

MTS4EA saves Trace files with a `.vpt` file extension.

Currently, the `.vpt` file is in fact a standard ASCII file which could be viewed in any text file viewer. The `.vpt` extension is used:

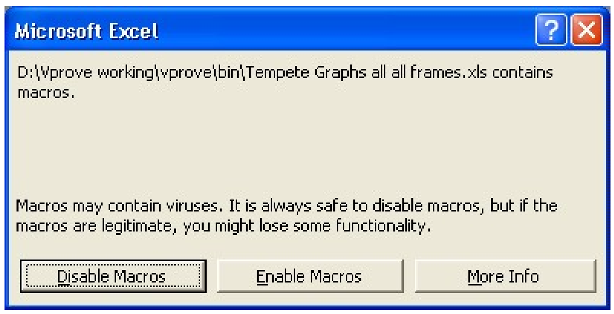
- So that this file extension is associated with MTS4EA in Windows, and by double-clicking on a `.vpt` file in Windows Explorer, the file is automatically opened in MTS4EA

View graphs…

This option calls Microsoft Excel to display the graphs generated: a macro is then run within Excel to create the graphs.

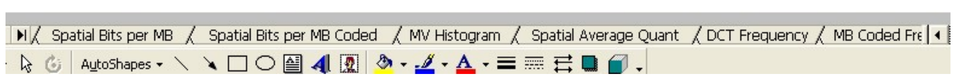
NOTE. *Macros must be enabled within Microsoft Excel to see the graphs.*

The default selection within Microsoft Excel is (usually) to disable macros (as below) - you must click on Enable Macros:

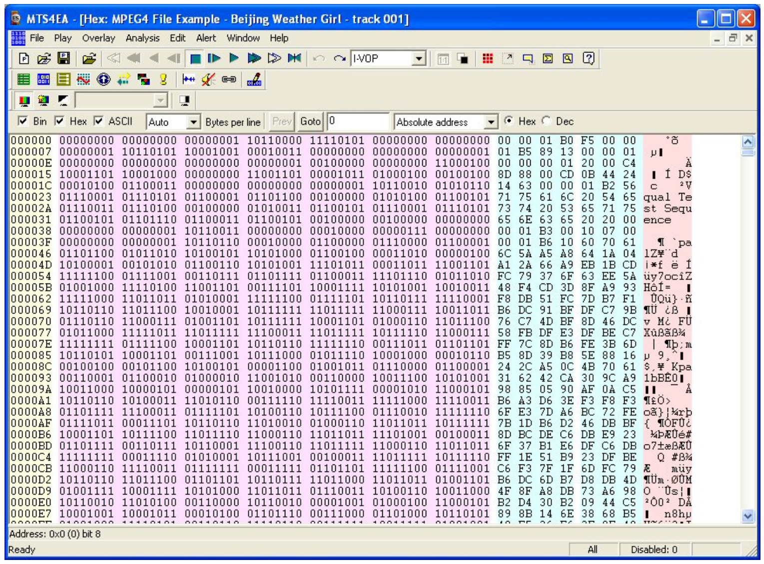


NOTE. *The data used to generate the graphs is available on the Data tab in the Excel file.*

The statistics options selected on the Graph enable tab within MTS4EA are then displayed on individual tabs within Excel:



View stream hex... `Ctrl+H`



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.*

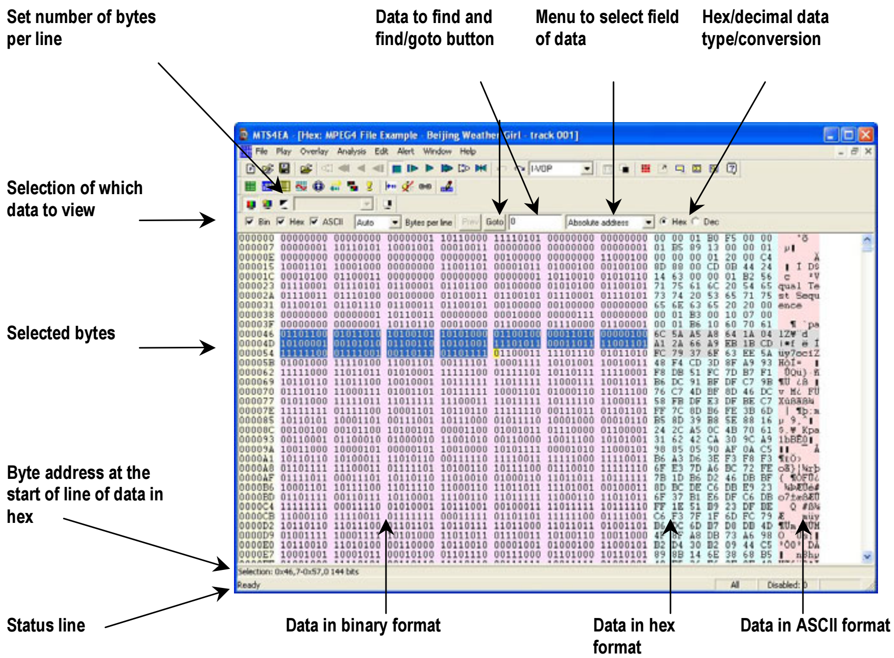
Using HexView, you can scroll through the file, and search for specific:

- 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



Setting information displayed; window width

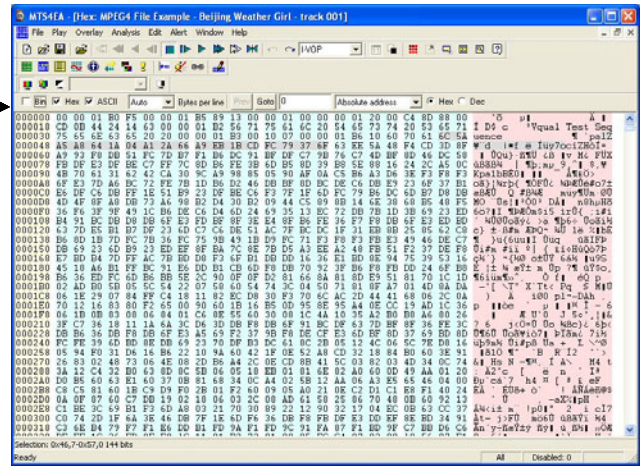
The width, size, etc. of the HexView window can be set as with any other window in Windows.

The same data is shown in each of the three sections, in their respective formats:

- Binary
- Hexadecimal
- ASCII

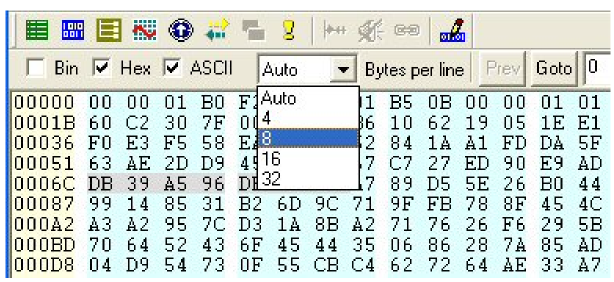
Each of these areas can be individually displayed/not displayed by clicking the appropriate check box in the top left.

In this example, the binary display has been switched off, leaving only hex and ASCII

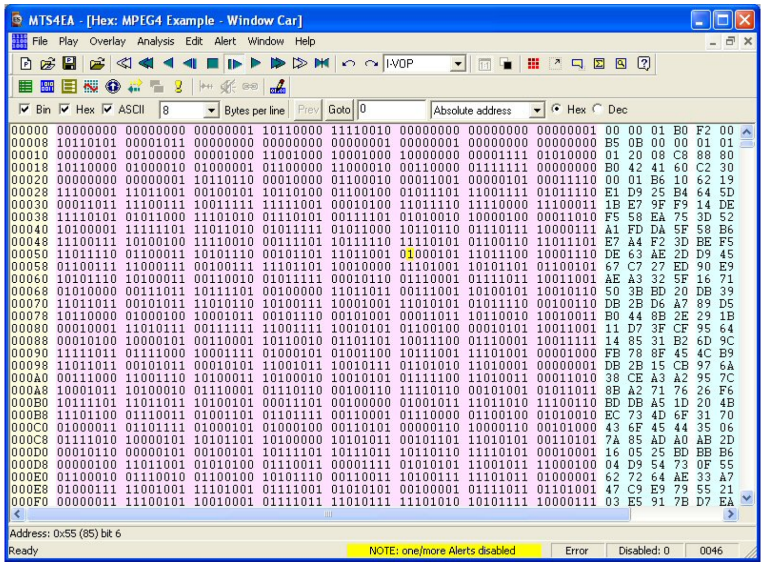


Setting bytes per line

This menu is used to set the number of bytes per line:

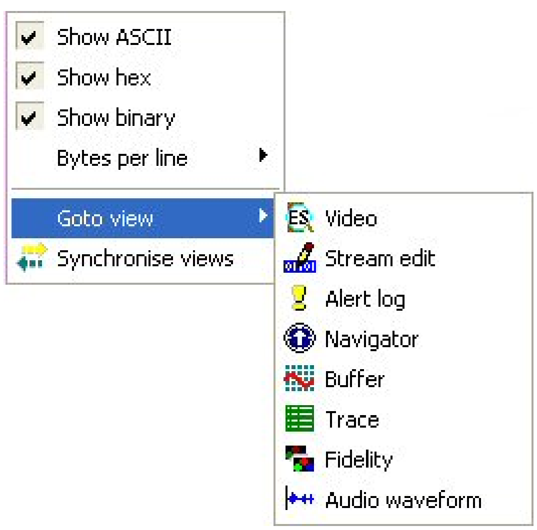


If the current width of the window is insufficient to show all the columns, then a scroll bar automatically appears at the bottom, allowing scrolling left/right to see all the columns, as in the following example:



Right-click pop-up menu/Goto view

Right-clicking in the HexView window generates the menu:



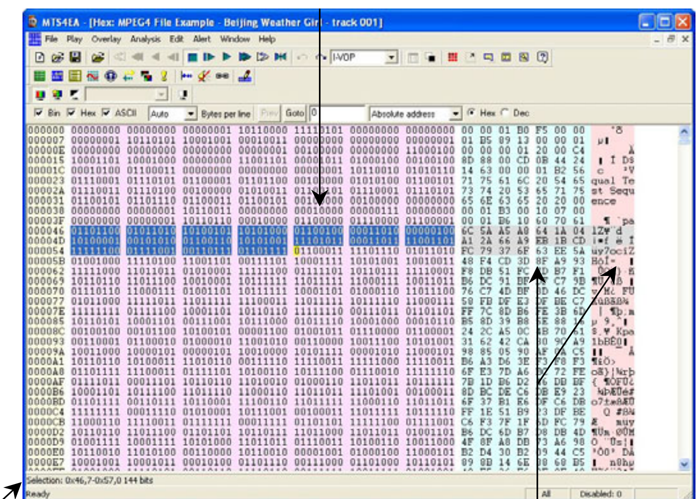
The top four buttons of the menu have the same functions as the buttons given on the top line (and as the HexView settings... option on the Window menu).

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:



The addresses of the selected area and the number of bytes selected are displayed in the status line

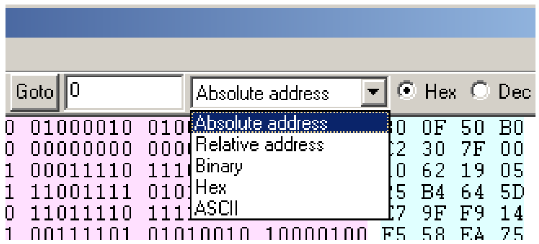
Whichever section is highlighted, the corresponding areas of the other sections are similarly highlighted

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 (e.g. 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 (e.g. entering anything other than 0-9 or a-f for a hexadecimal address) then 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.

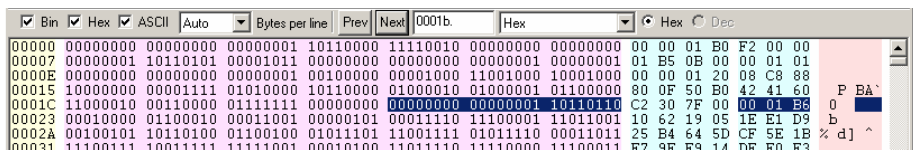
Wildcard searching using .

A wildcard can be entered in the data to be found - **.** [period].

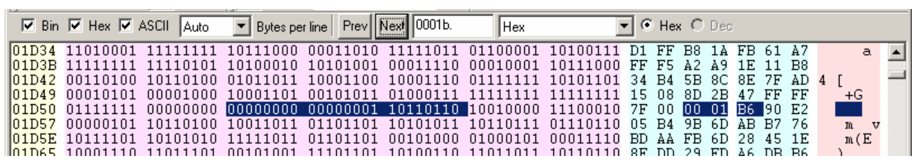
The wildcard matches a single digit in the base selected; the wildcard matches:

- 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 will find the following patterns in the selected example stream:



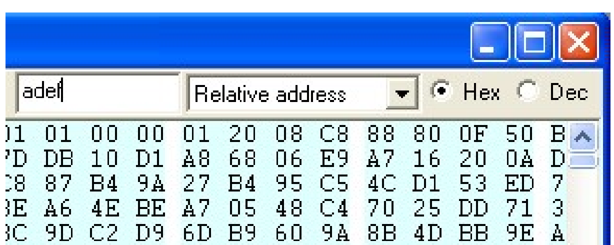
And



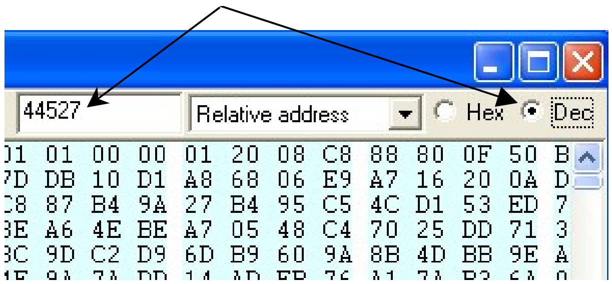
The wildcard can be inserted in any position, e.g. 03..456.8 is a valid search string, which will find any 9 consecutive digits where the first two are 03, digits 5-7 are 456 and the last digit is 8.

Conversion of hex<->decimal

If a hex value is entered into the find box:



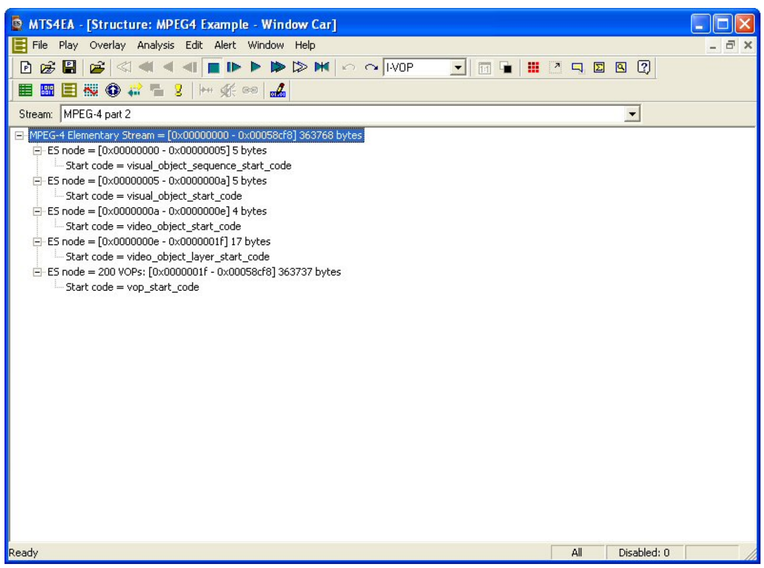
When the Dec button is clicked, this value is converted to hex (and vice versa):



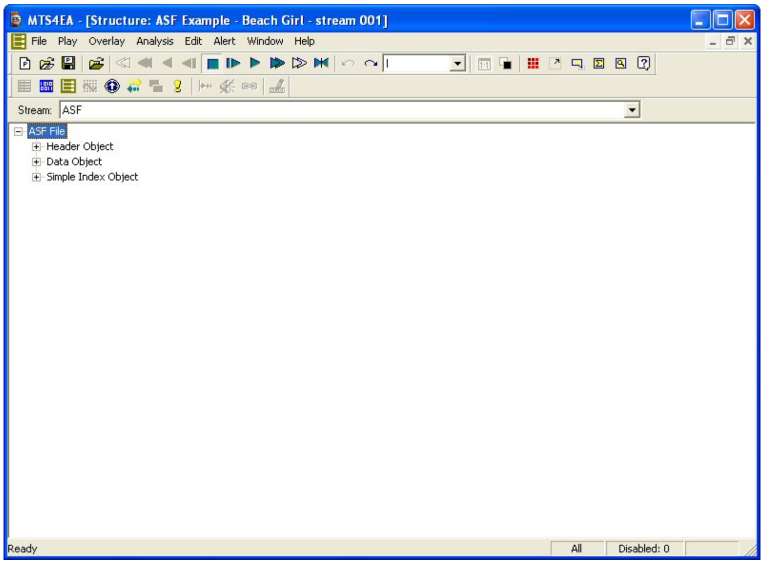
View file structure... Ctrl+R

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):

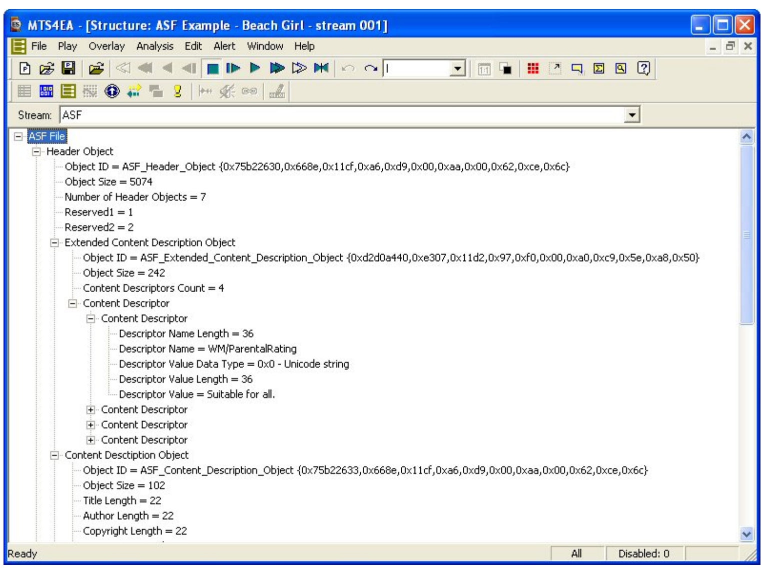
- MPEG-4 Elementary Streams:



- Container files, e.g. MP4, 3GPP, MPEG-2 PS and TS, ASF files:



- Each branch of the structure can be clicked to expand it and see the levels below:



View 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)

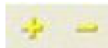
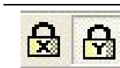
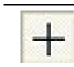
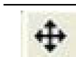
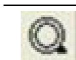
For MPEG-2 the analysis is of:

- 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 6–161 to **Buffer** analysis alerts/scroll bar area on page 6–165 (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 6–166 to *Buffer* analysis pop-up alerts: MPEG-4 and MPEG-2 on page 6–168
- The H.264/AVC HRD analysis is distinct and is explained in the sections beginning **HRD** buffer analysis: H.264/AVC on page 6–168 to **HRD** buffer overflow/underflow indication: H.264/AVC on page 6–170

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
	Lock X/Y zoom in/out and scrolling/panning. E.g. when the Lock Y button is pressed, zoom in and zoom out and scroll/pan only affect the X-direction. This allows (for example) the user 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

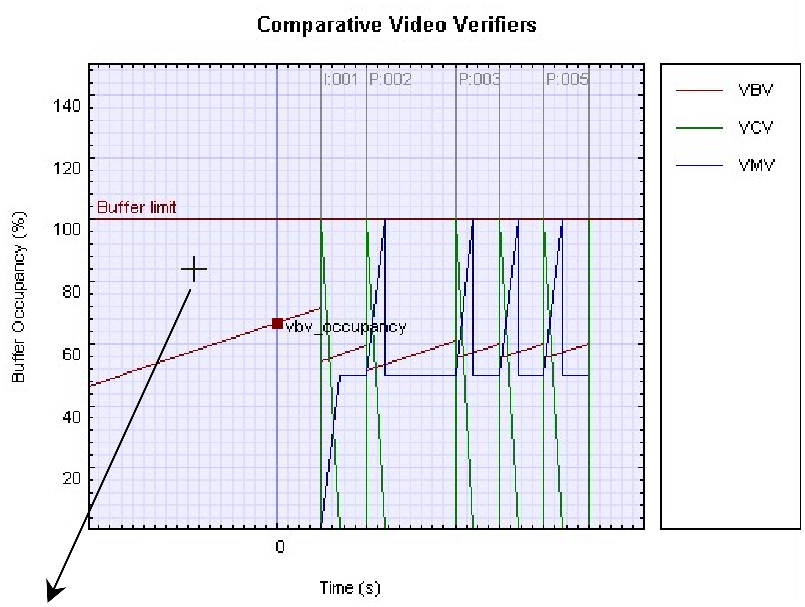
Right-click pop-up menu/Goto view

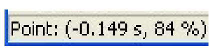
Icon	Equivalent toolbar icon	Function
Zoom-in		Zoom in (centered on current window)
Zoom-out		Zoom out (centered on current window)
Fit to range		Fit all data into the visible window
Goto origin		Go to the origin (time = 0)
X-Axis lock		Lock the X-axis when zooming/scrolling
Y-Axis lock		Lock the Y-axis 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	

The functions listed above from Zoom-in to Zoom tool inclusive are all explained in the previous section.

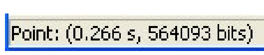
Using the measure tool 

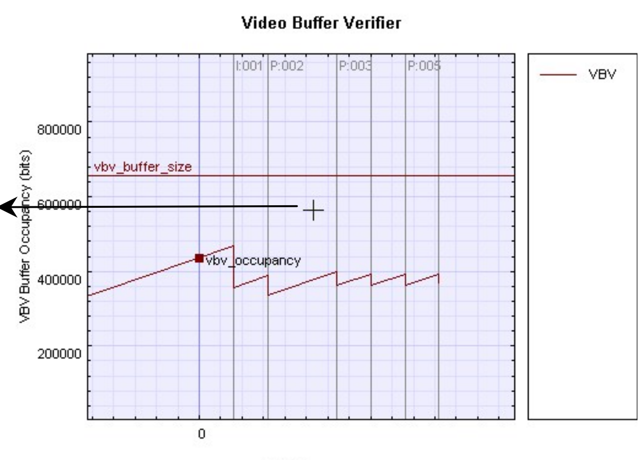
Moving the measure tool over the graph displays the values in the status line:



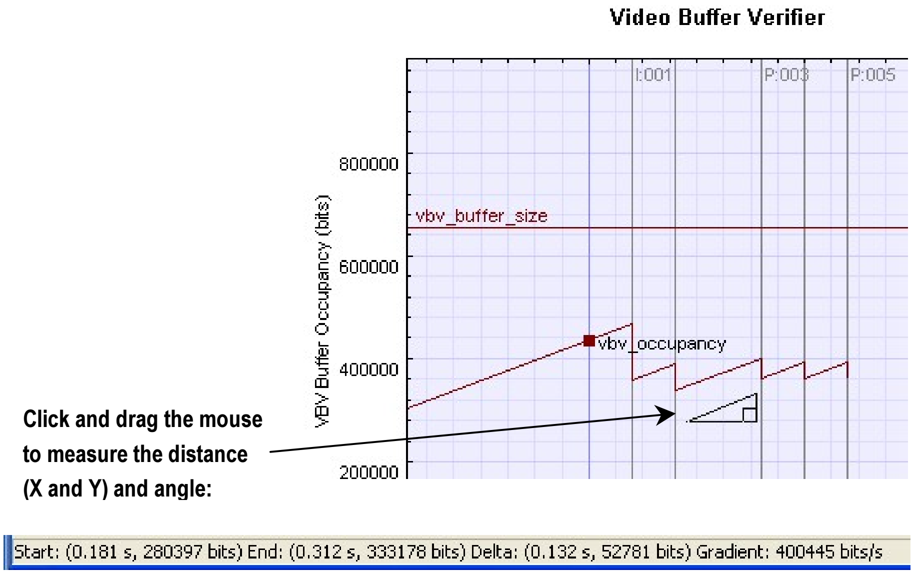


The left-hand value is the X-axis (time) value in seconds, to the nearest millisecond; the right-hand is the Y-axis value: either in percentage terms, if there is more than one graph; or in the correct units, if there is only one graph (as below)

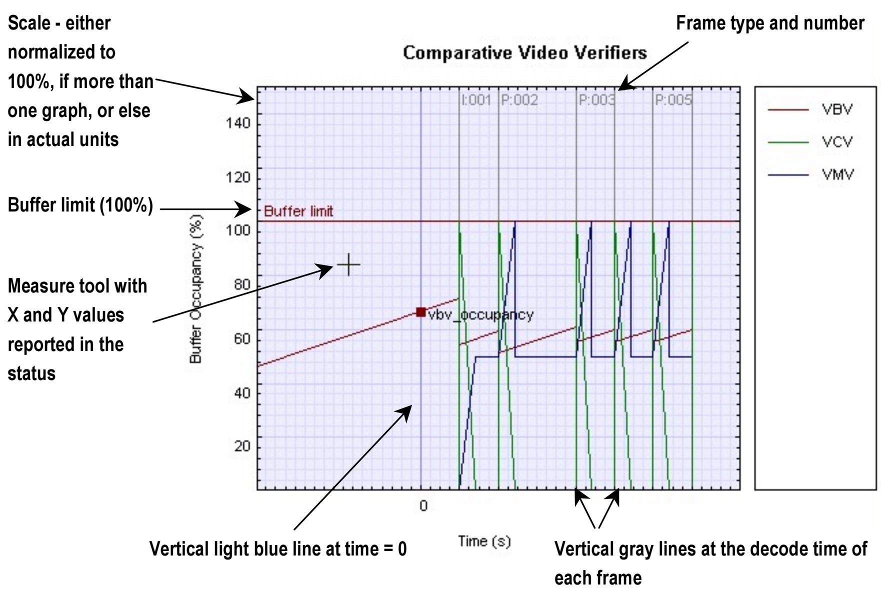




(Only the VBV is shown here)



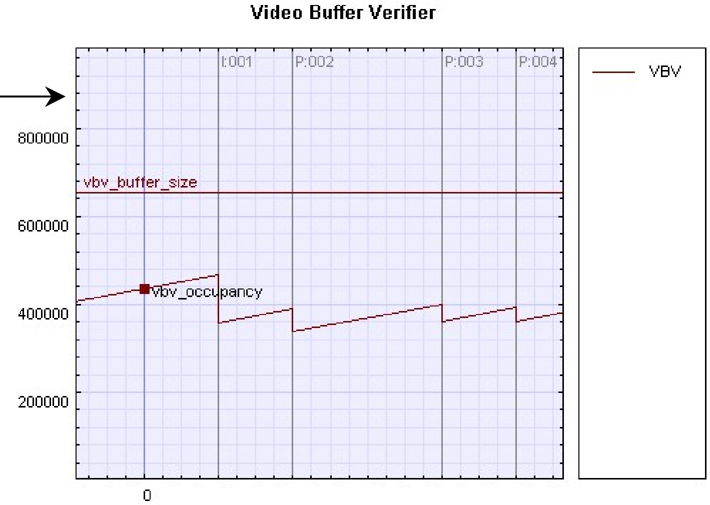
Buffer analysis graph area



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)



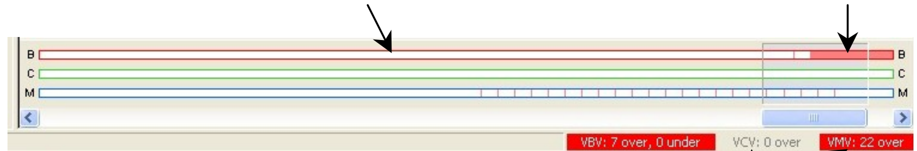
When the buffer data is too wide/too high for the current window, then scroll bars appear at the bottom/right (as appropriate).

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)

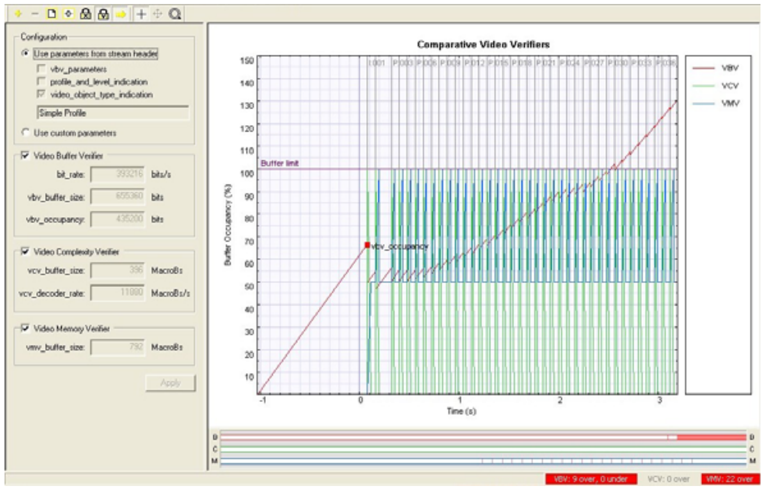
Red shows overflow; yellow shows underflow



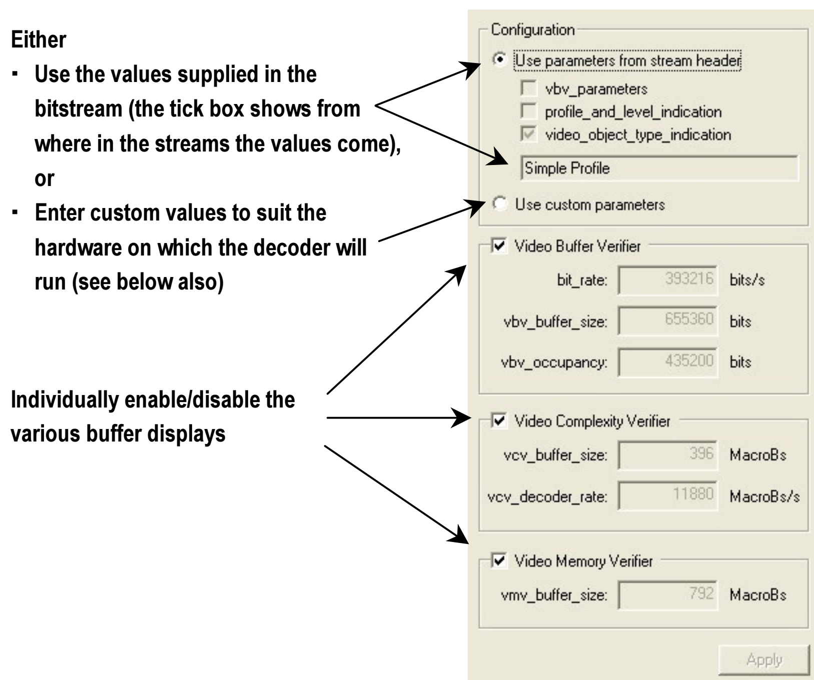
Each buffer item changes to red or yellow if there has been any overflow/underflow, and the number indicates in how many frames there is non-conformance.
If there is both overflow and underflow, then the color is red

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



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` and 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.

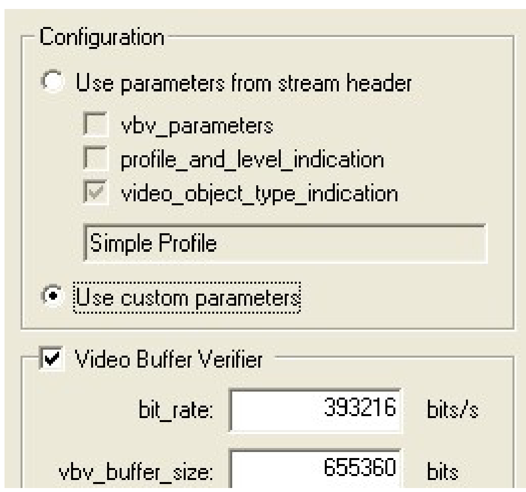
NOTE. *There is an order of precedence: if the* `vbv_parameters` *are specified in the bitstream then these values are used; if not, then the maximum values are used as allowed in the Profile/Level given by the* `profile_and_level_indication`*; but if there is no* `profile_and_level_indication` *then 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 MTS4EA will assume the maximum allowable values for the Profile/Level and these may be too large for the situation concerned
- 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

In this situation, the custom parameters are entered as below:



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 pop-up 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 pop-up 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, then pop-up alerts occur during video decoding unless disabled in the Alerts menu (the video must be playing for the pop-up alerts to be generated).

***NOTE.** Where the buffer parameters are implied by* `video_object_type_indication` *then no pop-up alert is generated in the event of exceeding the limits and nor is there a warning the in 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, it is essential that the user reads carefully and understands 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 (and, for example, only the Bus Junction and Grenadier Guards H.264/AVC example streams provided with MTS4EA have HRD information).

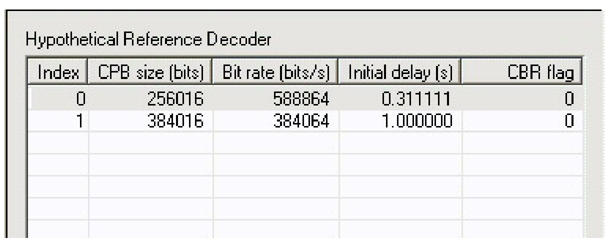
NOTE. *Where HRD parameters are not provided in an H.264/AVC bitstream, the icon and menu item for View buffer analysis... is grayed out.*

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 example below, there are two indexes/buckets: index 0 is selected and is used for the graph display, as shown in the screenshot below. 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). This is the size in bits of the Coded Picture Buffer (CPB).

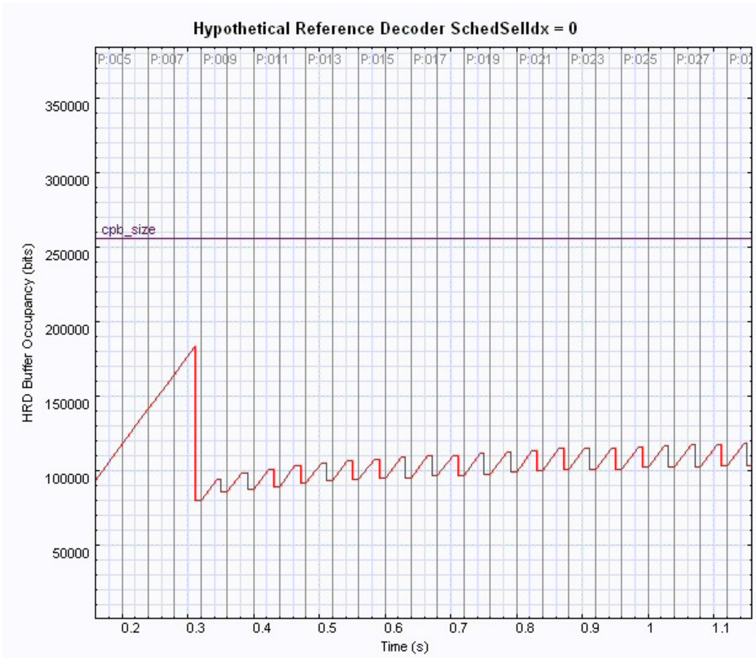
Bit rate (bits/s). The bit rate of the hypothetical transmission channel

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, then clicking on the appropriate index/bucket number displays the graph for that index/bucket (in the example below, index/bucket number 0):



The areas of the graph display are explained in **Buffer** analysis graph area on page 6–164 and **Buffer** analysis alerts/scroll bar area on page 6–165.

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:

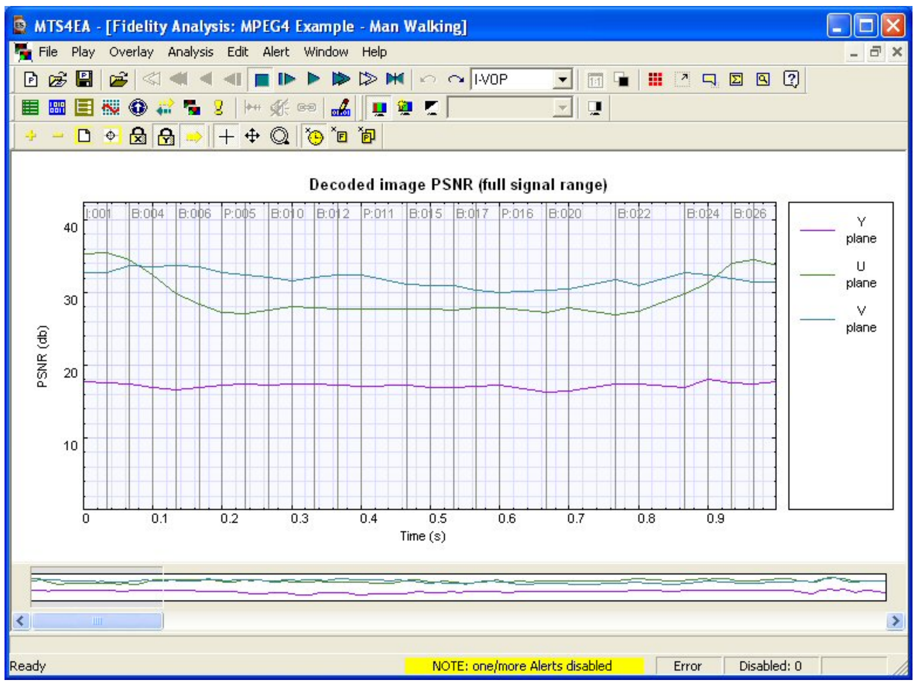
- In the bar line below the graph area
- In the status bar and
- In pop-up alerts

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 6–165).

The pop-up alerts for HRD buffer overflow/underflow are shown on-screen and enabled/disabled as with any other pop-up alert (see *Explanation of Fatal/Error/Warning/Info display* on page 6–180).

View 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:

- It has the same control icons – see **Buffer** analysis toolbar icons on page 6–161
- The graph area display is similar, with the same method display of frame types, frame numbers, frame times – see **Buffer** analysis graph area on page 6–164
- With the same means of measuring angles – see **Using** the measure tool  on page 6–163

- And the same right-click pop-up menu – see **Right-click** pop-up menu/Goto view on page 6–162

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 is 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.

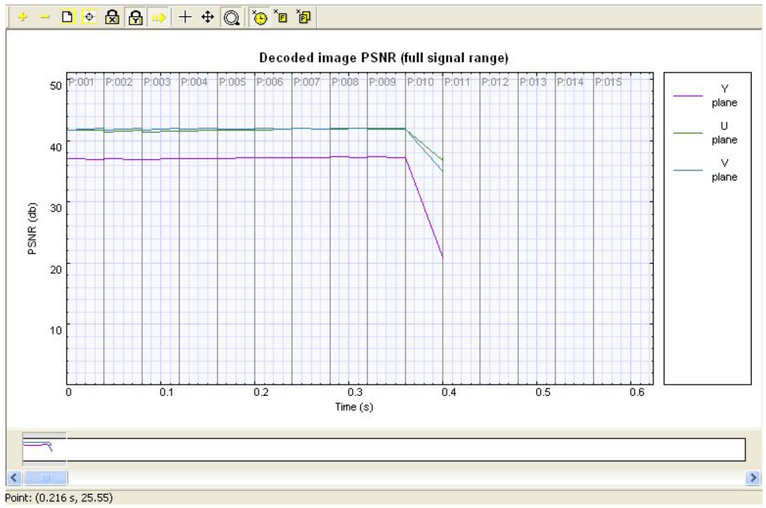
Fidelity analysis view icons toolbar

The icons above have the same functions as the icons on the buffer analysis toolbar; see **Buffer** analysis toolbar icons on page 6–161. The icons below are unique to the Fidelity analysis view:

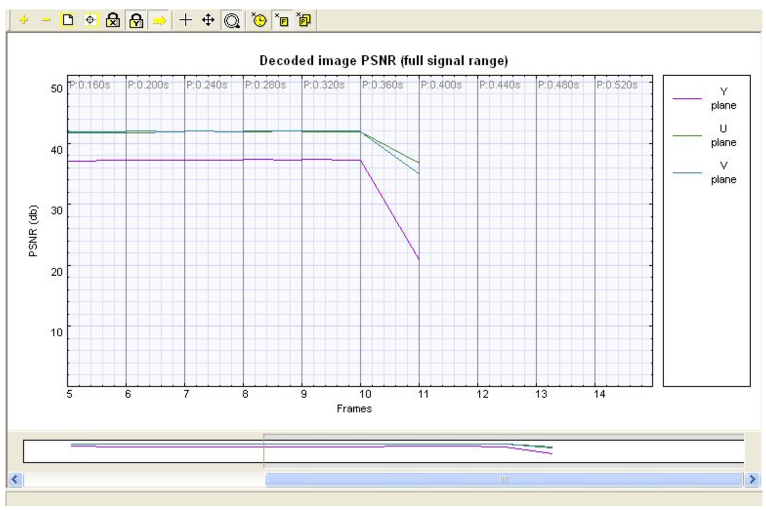
Of these three, the first two are available at all times. The third is only available when interlacing is in use and the picture can be broken down into units; it is grayed out at all other times.

These icons represent the following:

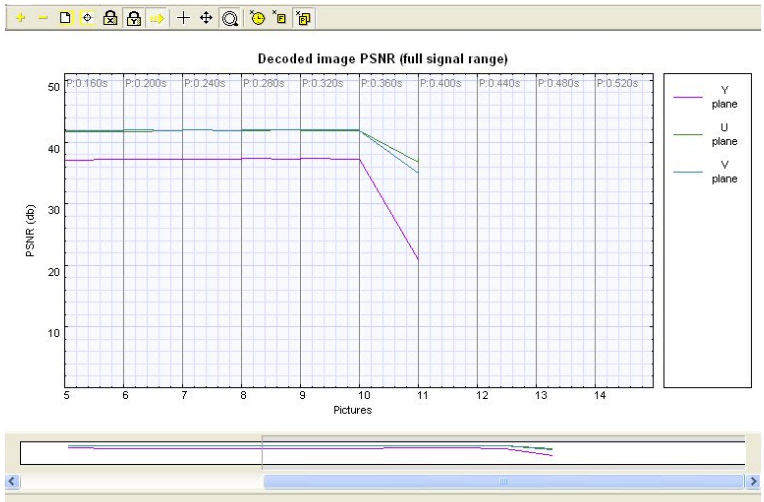
-  The ability to plot time in seconds on the x-axis, hence:



-  The ability to plot frame number on the x-axis, hence:



-  The ability to plot picture units on the x-axis, hence:



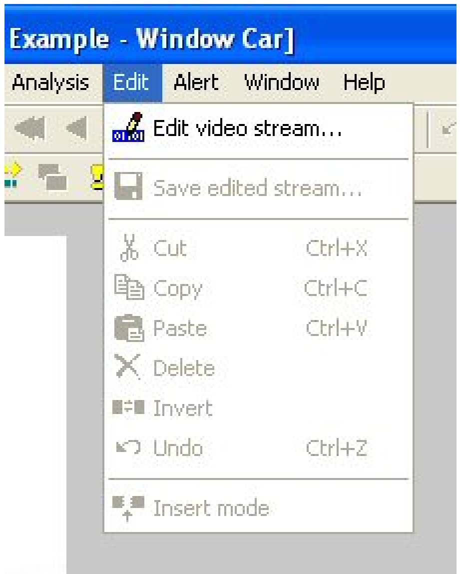
Fidelity metrics available

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 6–142.

Edit Menu

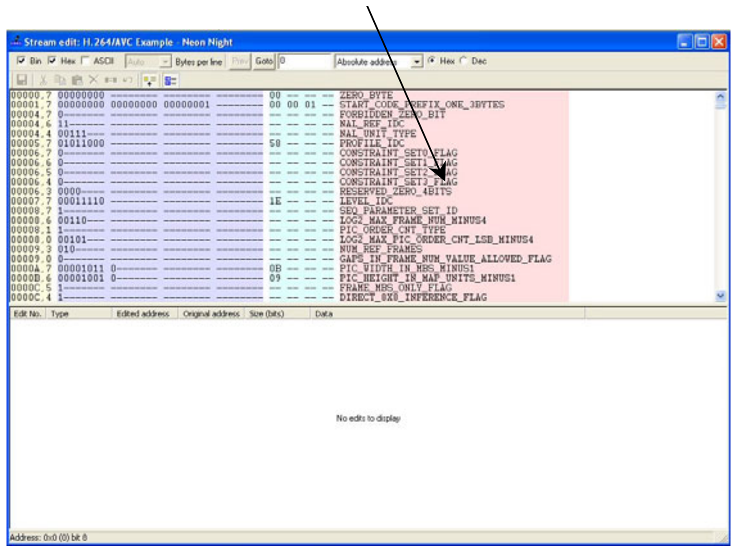


- The Bitstream editor facility allows the user to:
  - Insert, delete and modify bits and bytes at any point in the bitstream
  - Copy sections of the bitstream
  - Rerun the analysis on the edited bitstream
  - See a change log of all edits made

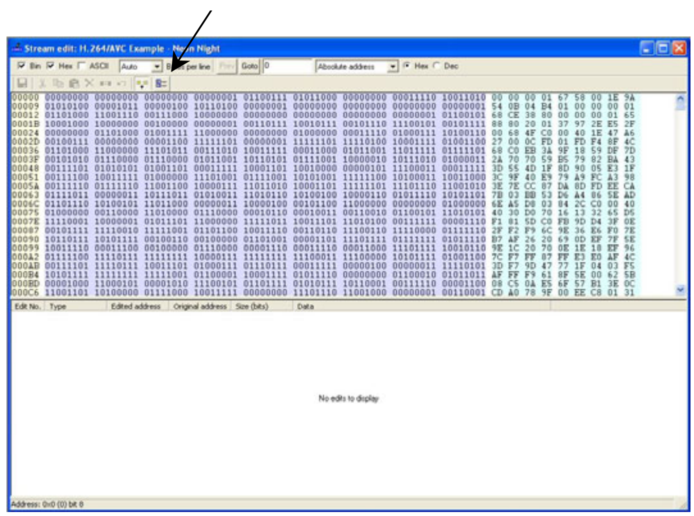
Edit video stream... 

When the Bitstream editor window is initially accessed, the following message will pop up advising the user that the videos syntax will not be visible in this view until it is played (decoded):

Once decoded, the syntax area will be populated.



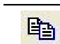
The pink area containing the syntax is displayed by default. If so desired, these labels can be removed from the display by clicking on this button:



The bottom half of the window - the change log - is empty until edits have been made to the stream.

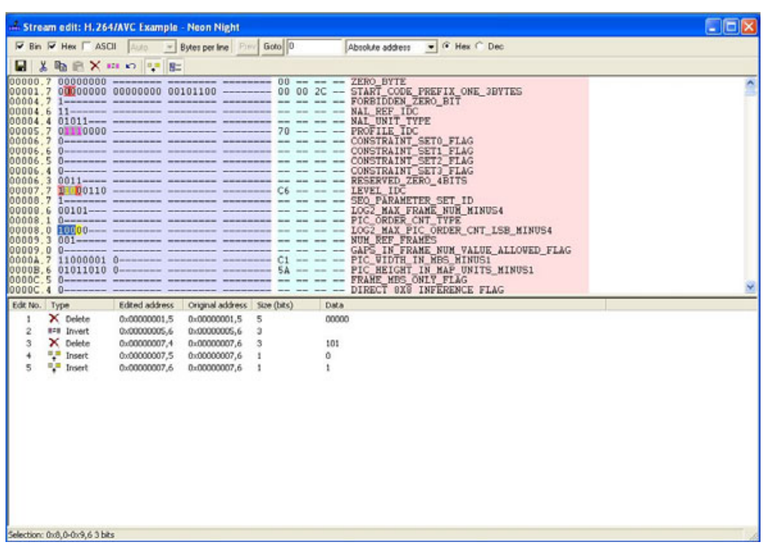
Bitstream editor icons

The functions of the remaining icons in the Bitstream editor view are outlined in the table below.

Icon	Function
	The edited stream can be given a new name and saved
	Cut (Ctrl+X) works as one would expect; only becomes available when part of the stream is highlighted
	Copy (Ctrl+C) works as one would expect; only becomes available when part of the stream is highlighted
	Paste (Ctrl+V) works as one would expect; only becomes available when part of the stream is highlighted and copied to the clipboard
	Delete bits; only becomes available when part of the stream is highlighted
	Inverts bits highlighted 0 becomes 1, 1 becomes 0; only becomes available when part of the stream is highlighted
	Undo last edit (Ctrl+Z)
	Insert bits mode (this is enabled by default)
	Show labels (this is enabled by default - see above)

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.



As can be seen above, against each numbered change listed are written both the Edited and Original addresses. This is because edits will have a cumulative or knock-on 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 of course 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 re-analyzed. Alternatively, the stream may be saved under a new name and re-opened later for analysis.

Alert Menu

This menu controls the setting of the overall level of the pop-up alerts and gives access to configure individual alerts as enabled/disabled.

These pop-up alerts occur when the video is being decoded. Consequently:

- The pop-up 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 video stream, the pop-up alerts will not appear if that part of the bitstream has already been decoded and is in the step-back buffer cache (see *Pop-up alerts when seeking forwards/backwards* through data in the step-back buffer (cached) on page 6–183)

NOTE. *For some Alerts, extra information is provided about the section of the video that has generated the error, and the standard concerned if it is standard-specific. See* **General** *codes used in Trace files and Alerts on page 7–1 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 6–187 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 switched off).*

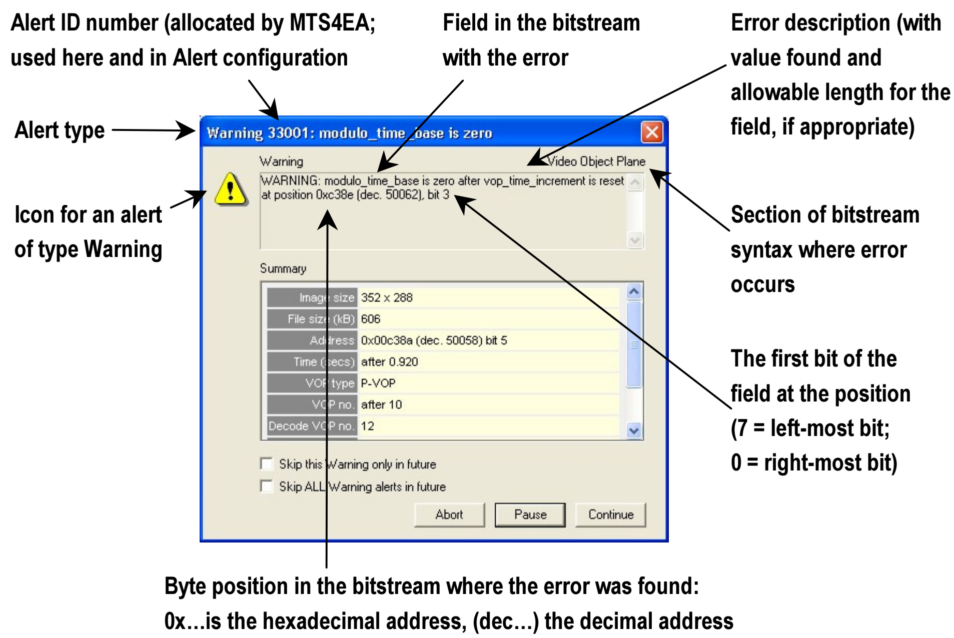
Each time a video stream is opened in MTS4EA, the alert level is set to Info (the strictest) and all Alerts are reenabled.

When an alert is triggered, a message box appears: at the same time, video decoding is suspended.

The message box will provide the user with the reason that the video decoding has been interrupted – see below.

Explanation of Fatal/Error/Warning/Info display

Information at top of alert pop-up



***NOTE.** For bitstreams which contain B-VOPs/B-frames, it is vital to understand that the pop-up 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 6–181 for more information.*

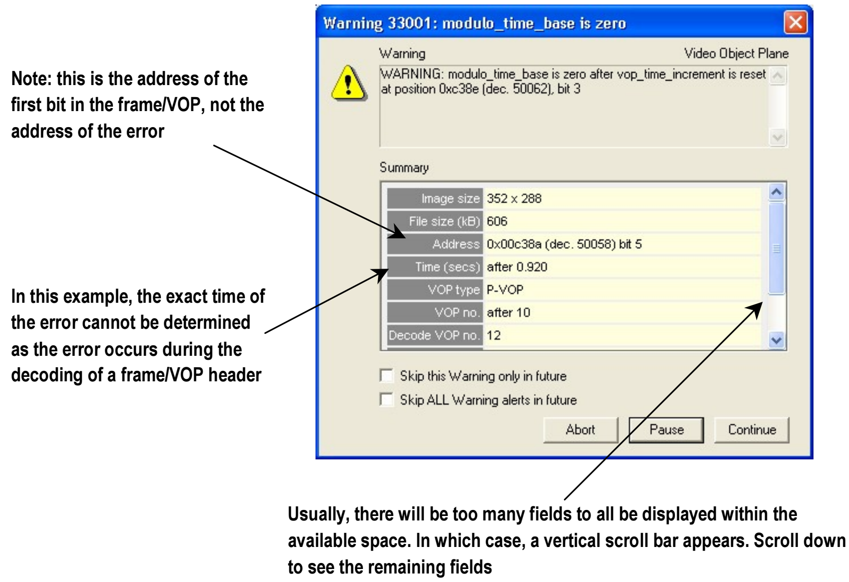
For all Fatal, Error, Warning and Info displays:
- the bitstream position of the error is the number of bytes from the beginning of the video bitstream, where the first byte is byte zero;
- the position is displayed in hexadecimal (0x...) and decimal (dec. ...);
- the bit start is the first bit of this 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 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 which 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 value given. See **Explanation** of bit/bit start on page 7–6 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 **Summary** Tooltip Ctrl+U on page 6–67 for a detailed explanation of the fields and provided data.



Decode/display frame/VOP numbers in bitstreams with B-frames/B-VOPs

The pop-up alert displayed relates to the Warnings, Errors, etc., that MTS4EA finds during decoding of the video.

This means that for bitstreams which contain B-frames/B-VOPs (where the decode order is different from the display order) it can appear that the pop-up 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

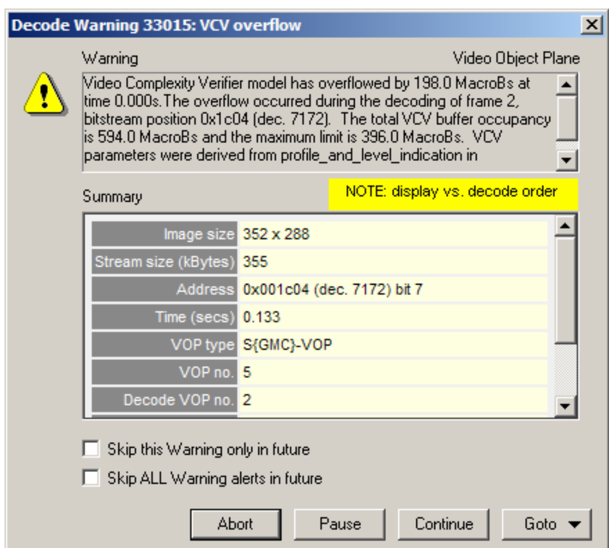
- Order of data in displayed:

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 the diagrams above, 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 pop-up alert:



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.*

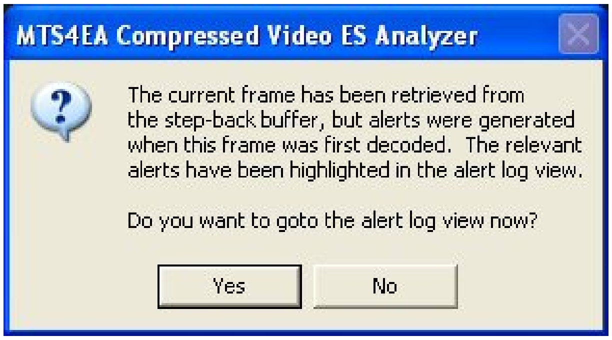
Pop-up alerts when seeking forwards/backwards through data in the step-back buffer (cached)

When the video is decoded, pop-up alerts may occur. The results of the decoding are stored in a cache, the Step-back buffer - see **General** on page 6–51 for more information.

It is possible to step forwards/backwards over the frames where these alerts occur. To avoid a continual repetition of the pop-ups (and potentially many such alerts appearing), the pop-up alerts are not displayed when seeking forwards/ backwards in these situations.

If one or more pop-up alerts would have occurred, then this is indicated in one of two ways:

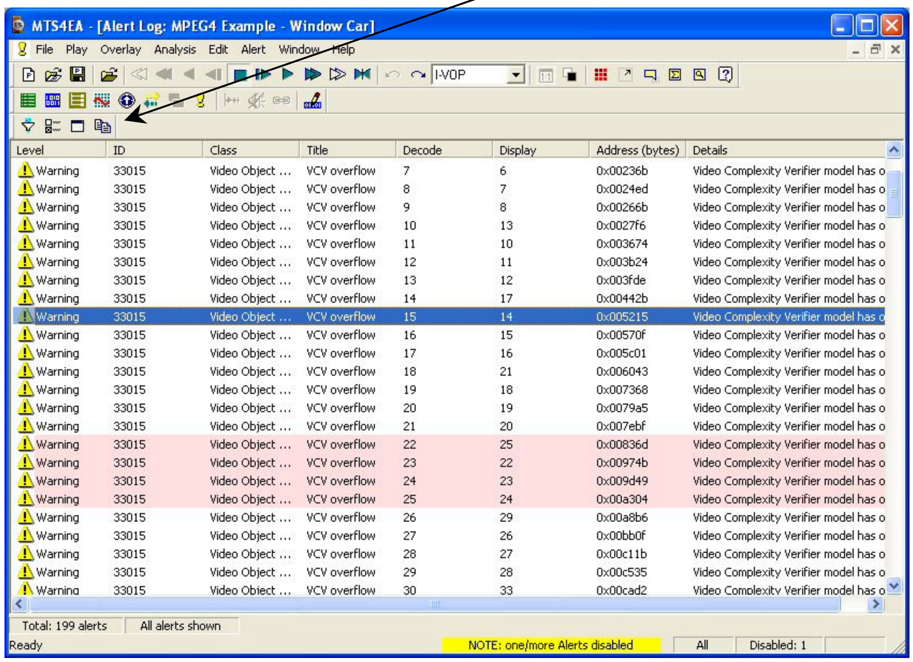
- There may be a pop-up message, at most once per seek forwards or backwards:



- If the Alert log window is open, then the alerts that would have popped-up are highlighted in the Alert log window (see **Alert** log  on page 6–191 for more information)

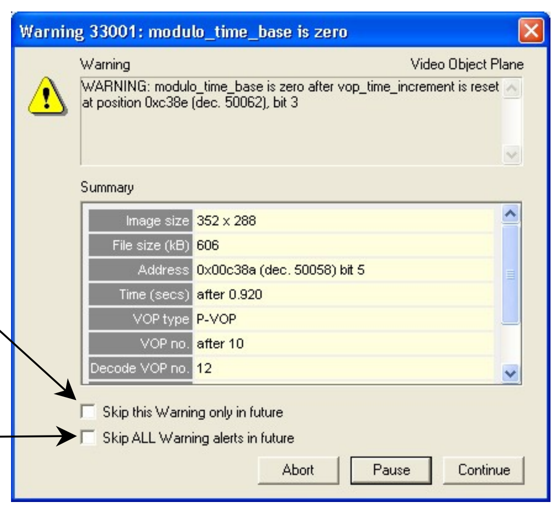
Copy alert details to the Windows clipboard

The details of an alert can be copied using `Ctrl+C` or the copy icon.



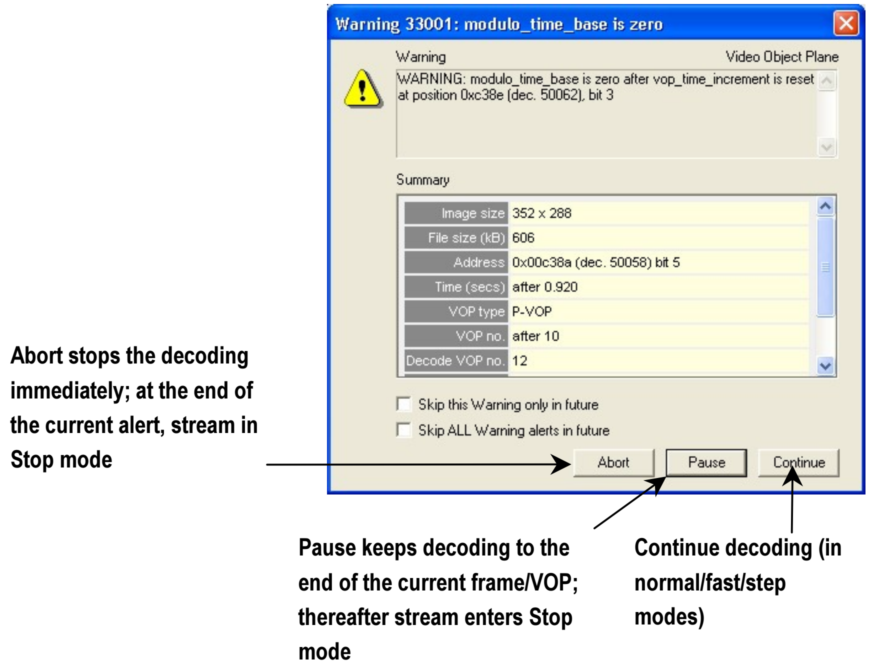
Check box options

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

Abort, Pause, Continue



NOTE. *Abort can also be done 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 which 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 pop-ups 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 (PSNR, etc.)	
Audio waveform	Graph of audio stream: Y-axis is normalized amplitude; X-axis is time (seconds)	

Description of Alert levels

Fatal

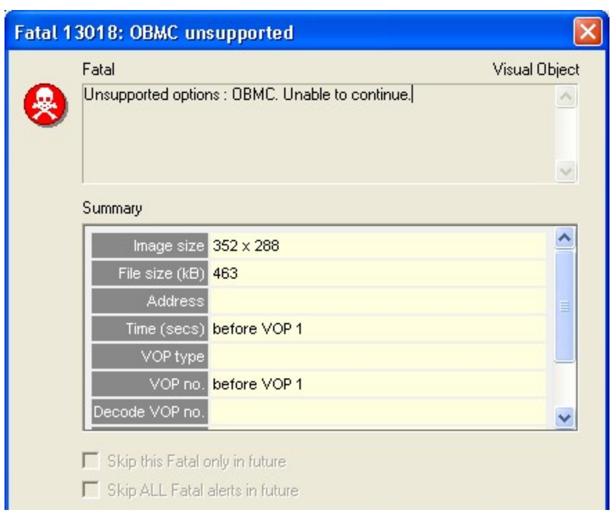
A Fatal error occurs when there is something encountered in the bitstream that cannot be decoded by MTS4EA.

This selection cannot be switched off, and of course, the decoder cannot continue after a Fatal error.

Fatal errors in 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

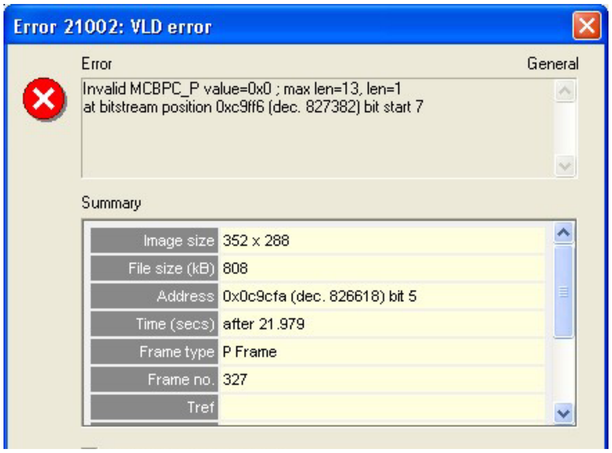
For example, the Fatal error below was caused by the OBMC flag in an MPEG-4 bitstream being set (on), that the video bitstream in question is using overlapped block motion compensation. (OBMC = Overlapped Block Motion Compensation: although it is in the current MPEG-4 standard, none of the MPEG-4 profiles or levels defined currently support OBMC.)



Error

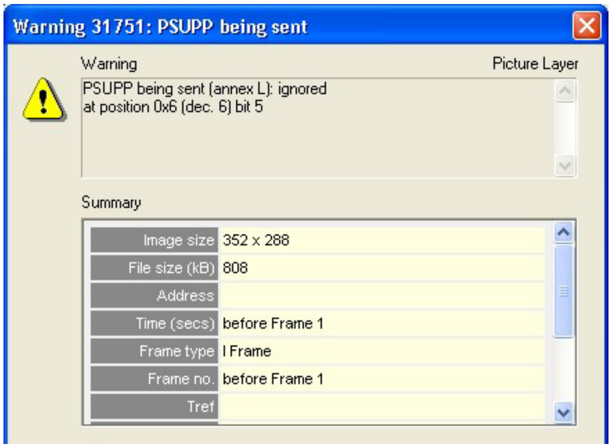
When this item is selected, the decoder will stop at any point where there is an Error in the bitstream.

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):



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 give rise to such warnings (the example below in an H.263 stream):



Info

When checked, this item will provide information pop-ups 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.

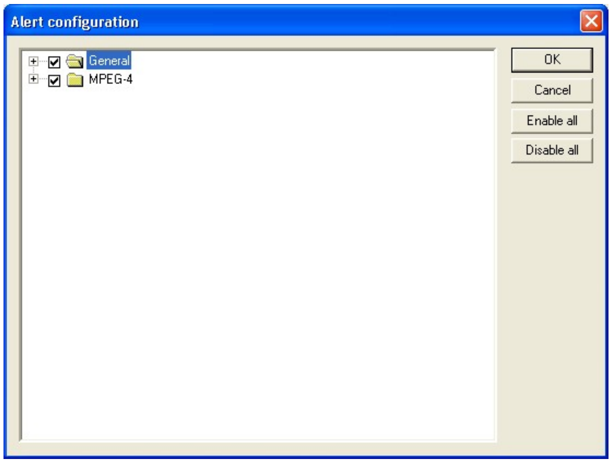
NOTE. *The overall alert level is set from the Alerts menu options Errors, Warnings and Infos, and this the enabling of specific alerts but not the disabling of specific alerts.*

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 6–198
- 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 6–198

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.*

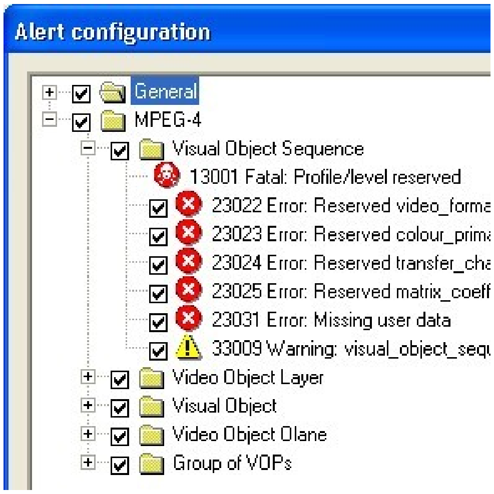
Enable/disable specific alerts



This displays two folders at the top level:

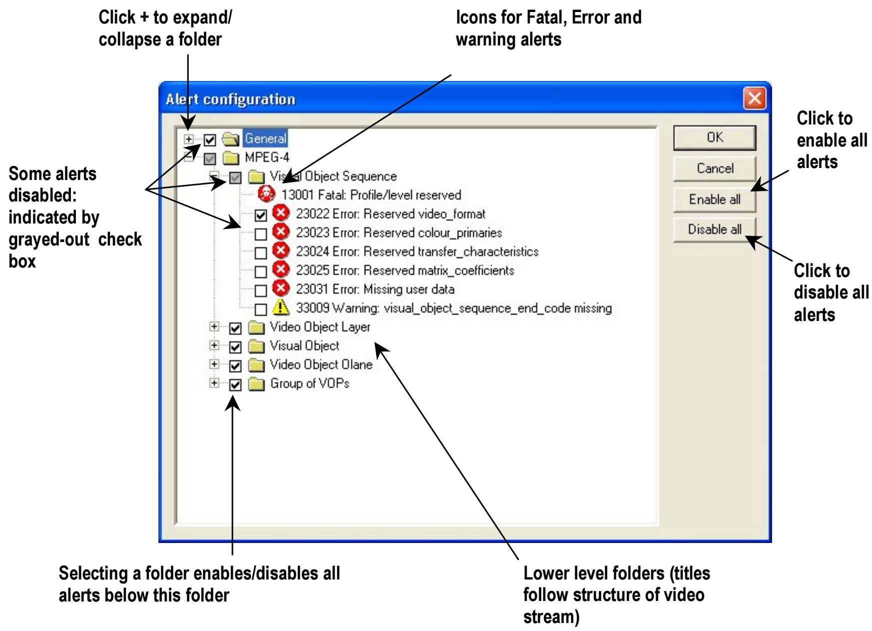
- General alerts, which can apply regardless of the standard concerned
- A standard folder (MPEG-4, H.263 or H.261) that contains alerts specific to the standard concerned

The + next to each folder can be clicked to reveal lower levels:



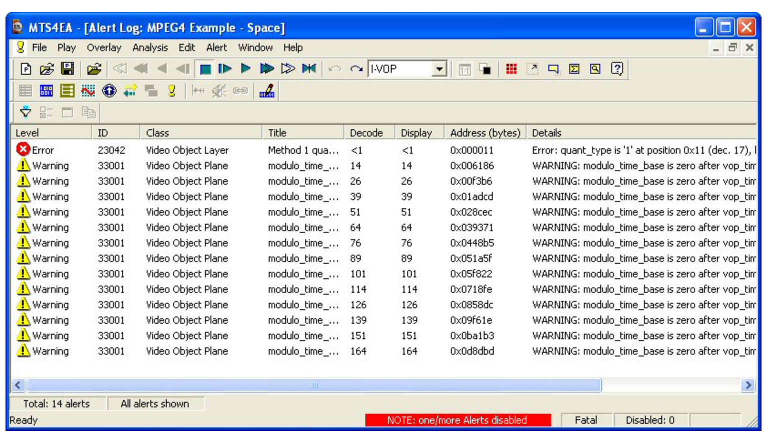
Elements of the Alert configuration window

NOTE. This is used to configure the individual alerts: note that even if an alert is enabled here, if the overall alert level is set higher then the alert will not trigger a pop-up.



Alert log

The Alert log keeps a list of the pop-up alerts that have occurred during the decoding of the bitstream concerned:



Alert log window icons & column titles

Alert filter icon See Showing/hiding **alerts (alert filter)** on page 6–193

Configure alerts icon Clicking on this icon opens the Alert configuration dialog box - see **Configure** Alerts on page 6–189.

Note that 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 6–194.

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 concerned: see the screenshot above which shows an error in the Video Object Layer and 7 warnings in the Video Object Plane within the MPEG-4 bitstream concerned.

Note that 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 has been found.

See *Decode/display frame/VOP* numbers in bitstreams with B-frames/B-VOPs on page 6–181 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.*

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 6–181 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 has been found.

NOTE. *For video bitstreams that have been extracted from container files (e.g. 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.*

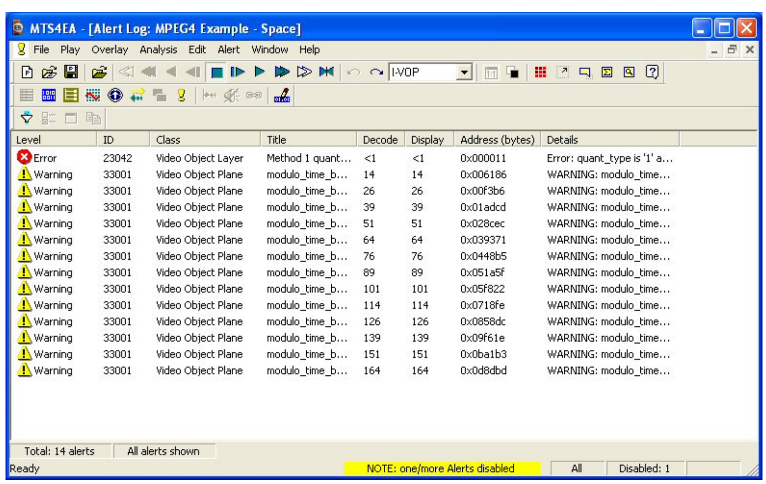
Details. This is the description text that is shown when the pop-up alert concerned is displayed.

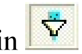
Showing/hiding alerts (alert filter)

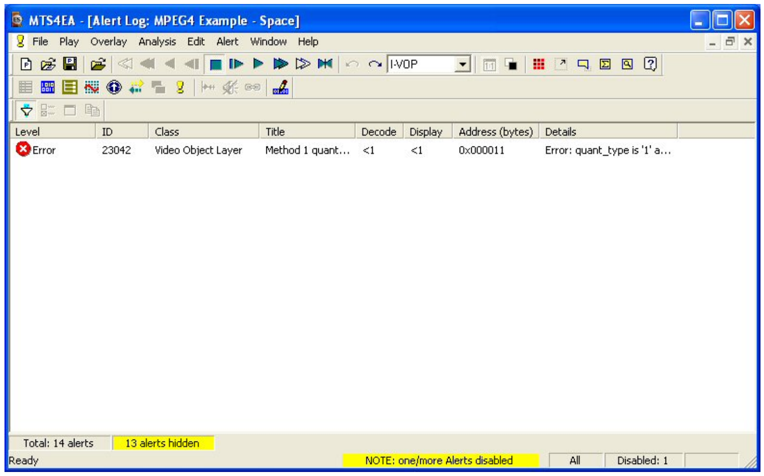
In some cases there can be hundreds or thousands of alerts. It can be that these are known issues and therefore the user does not want to list these in the Alert log window (as these would clutter the window unnecessarily).

Therefore the alerts that are shown in the Alert log window can be filtered: the filtering is based upon the setting of the alert pop-up enable in the Alert configuration (see **Configure** Alerts  on page 6–189 regarding the Alert configuration dialog box):

- If the filter icon is out  then all alerts are shown regardless of whether or not they are enabled for a pop-up alert:



- If the filter icon is pushed in  then only those alerts that are enabled are shown:



(In this case, 13 alerts are hidden out of the total of 14 alerts)

The Alert log filter can also be switched on/off by right-clicking and selecting the option from the pop-up menu.

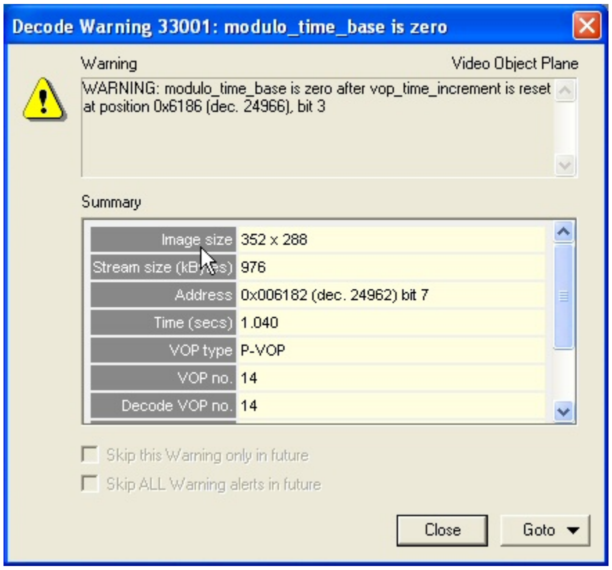
NOTE. *When the filter icon is grayed out, this is because all the alerts are enabled for display, so that there are no alerts to filter out (there are no alerts not to show).*

View details of the alerts

This can be done in a number of ways:

- Double-clicking an alert in the Alert log window
- Selecting an alert by single-clicking an alert, then right-click to get the pop-up menu, then click View alert details
- Selecting an alert by single-clicking an alert, then clicking the icon

When this is done, a display is shown very similar to the original pop-up alert display:



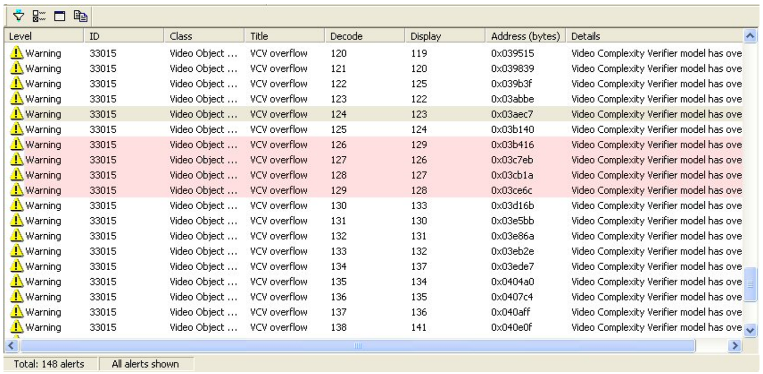
Alert highlighting when seeking forwards/backwards through video in the step-back buffer

MTS4EA buffers video (and the data associated with each video frame) in a step-back buffer or cache: see **General** on page 6–51 more information.

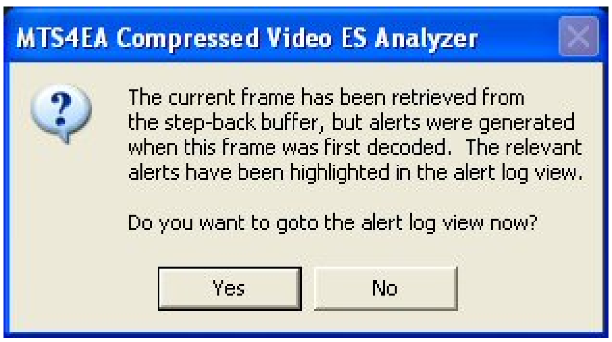
When stepping backwards/forwards through video that is held in the step-back buffer, to avoid continually having known alerts appearing (which then require to be cleared or skipped), MTS4EA simply highlights the alerts relevant to the area being skipped in the Alert log.

For example, if skipping forward 100 frames and there would normally four pop-up alerts during the decoding of the 100 frames, then MTS4EA will highlight the four alerts concerned in the Alert log (unless the alerts have been skipped and are therefore hidden in the Alert log, as indicated by the status bar of the Alert log).

An example of the highlighting of alerts is shown below:



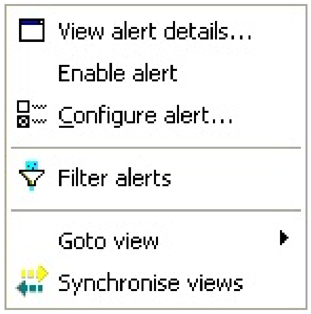
If the Alert log is closed, then MTS4EA displays a warning, as below:



This is displayed once only for all the Alerts between the relevant frames.

Right-click pop-up menu

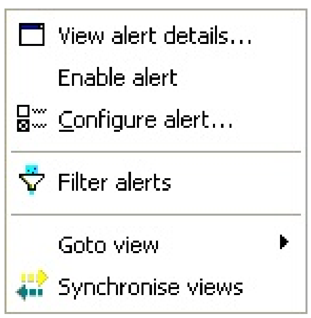
The functions available in the Alert log can be selected by right-clicking and selecting the function from the pop-up menu:



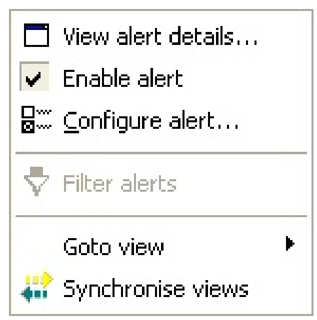
The functions of this menu are:

 **View alert details...** See View **details of the alerts** on page 6–194.

Enable alert. An individual alert can be enabled or disabled by clicking on this menu selection:



Alert disabled



Alert enabled

NOTE. *This allows the individual alert concerned to be enabled/disabled from this menu, without having to open the Alert configuration dialog box.*

 **Configure alert...** See Configure **Alerts**  on page 6–189.

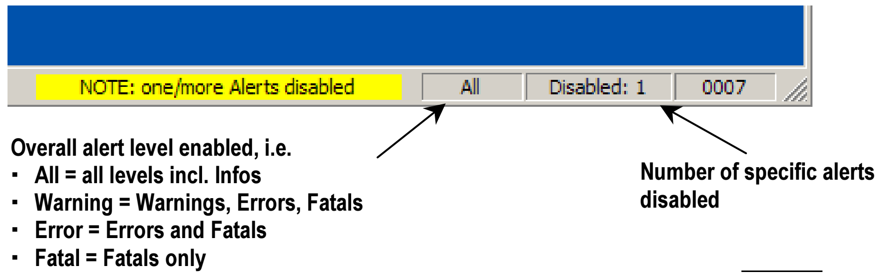
 **Filter alerts.** See Showing/hiding **alerts (alert filter)** on page 6–193.

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.

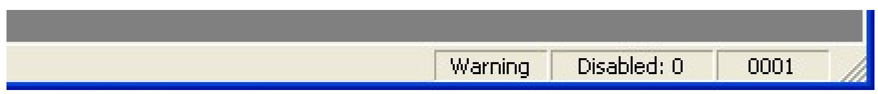


One or more alerts disabled

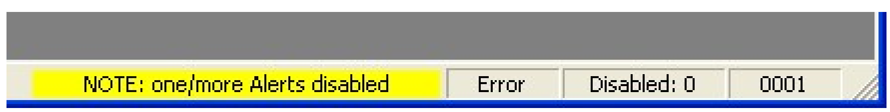
When one or more alerts are disabled, either because:

- The overall alert level does not include Warnings, Errors and Fatals
- One or more of the specific alerts have been disabled

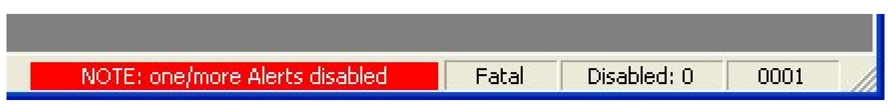
The status bar changes as below:



Overall alert level: Infos disabled (all other levels enabled)



Overall alert level: Warnings disabled (Errors and Fatals enabled)

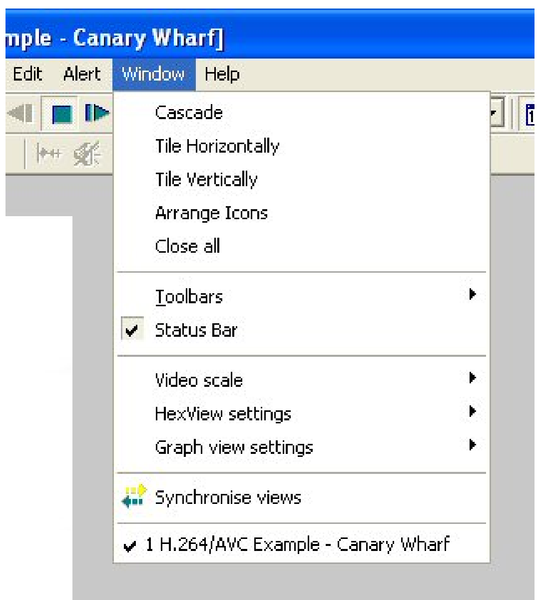


Overall alert level: Errors disabled

Overall alert level: All but one/more specific Warning alert disabled

Overall alert level: All but one/more specific Error alert disabled

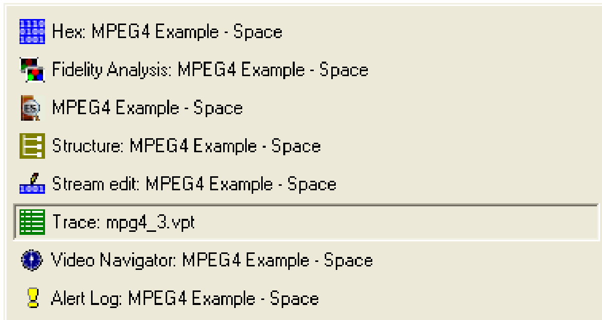
Window Menu



The commands on this menu comprise many commands typical of a standard Windows application, such as arranging windows; list of open windows.

Quickly switch windows

Pressing `Ctrl+Tab` together brings up a list of open MTS4EA windows:

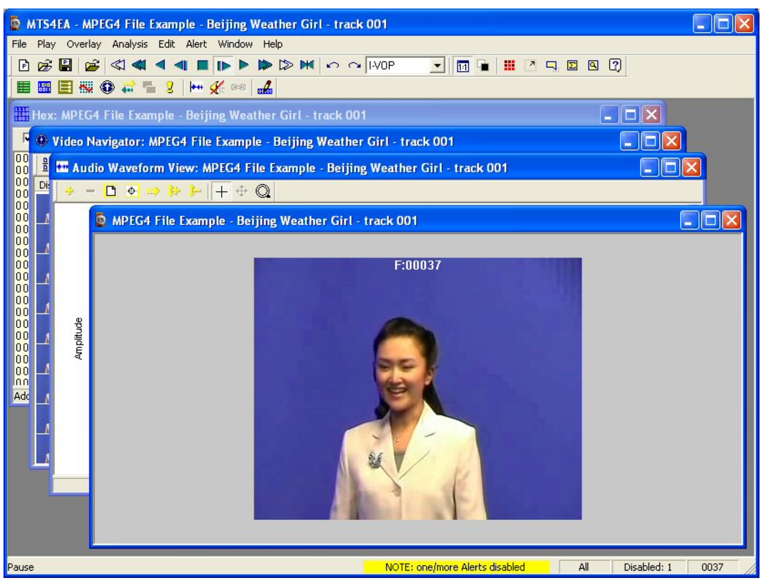


Repeatedly pressing `Ctrl+Tab` steps through all the open windows. Alternatively, the mouse can be used to select a window.

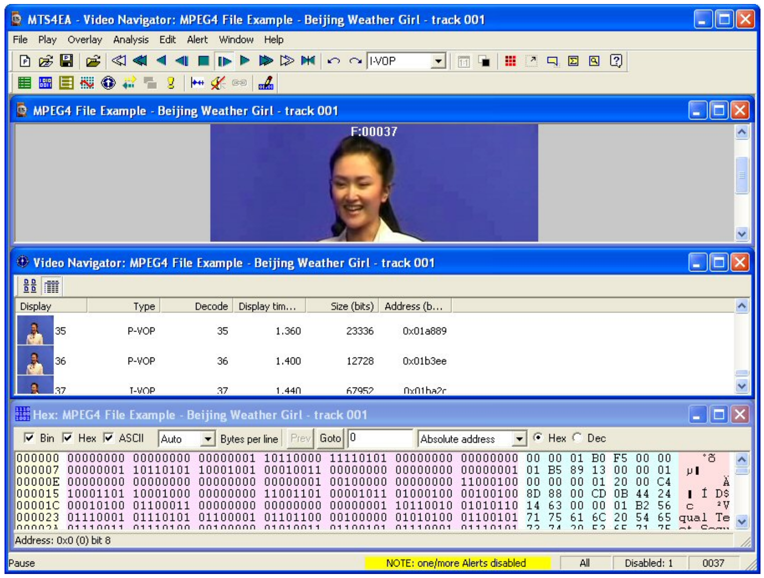
Cascade, Tile, Arrange Icons, Close All

These function as per standard Windows commands:

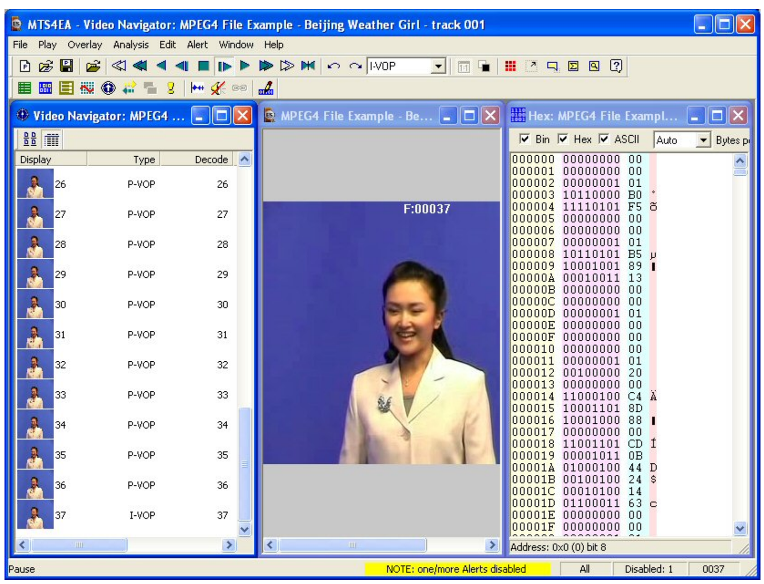
- Cascade arranges the open windows in a cascade:



- Tile Horizontally arranges the open windows like so:



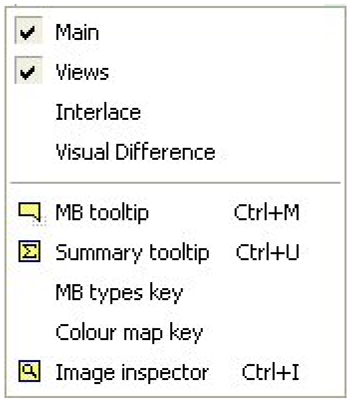
- Tile Vertically arranges the open windows like so:



- Arrange Icons lines up the windows when they are minimized
- Close all closes all windows in MTS4EA

Toolbars

In the example below, the Main and Views toolbars are shown, but not the Interlace and Difference toolbars:



Main

When selected (which is the default setting), the Main toolbar is displayed with the icons visible. This is indicated by the tick in the Window menu. When not selected, it is hidden.

At various times, some or many of the toolbar buttons are grayed out, indicating that they are disabled, when the particular button cannot be used as the function concerned cannot be active at the time.

See *Functions* of the Main toolbar buttons on page 6–211 for more information.

Views

The Views toolbar is shown by default. See *Functions* of the Views toolbar buttons on page 6–212 for more information.

Interlace

The Interlace toolbar is not shown by default: it is always switched 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 6–213 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 **Visual** difference on page 6–106 for more information.

MB tooltip. See MacroBlock **tooltip** `Ctrl+M` on page 6–76

Summary tooltip. See Summary **Tooltip** **Ctrl+U** on page 6–67

MB types key. See ***MacroBlock*** **Types** `Ctrl+Y` on page 6–86.

Color map key. See Colors on page 6–95.

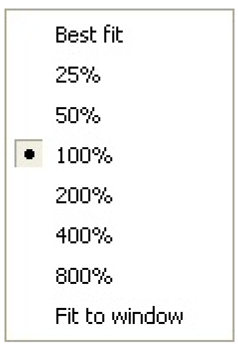
Image inspector. See Image **inspector** `Ctrl+I` on page 6–85.

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 6–216 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.

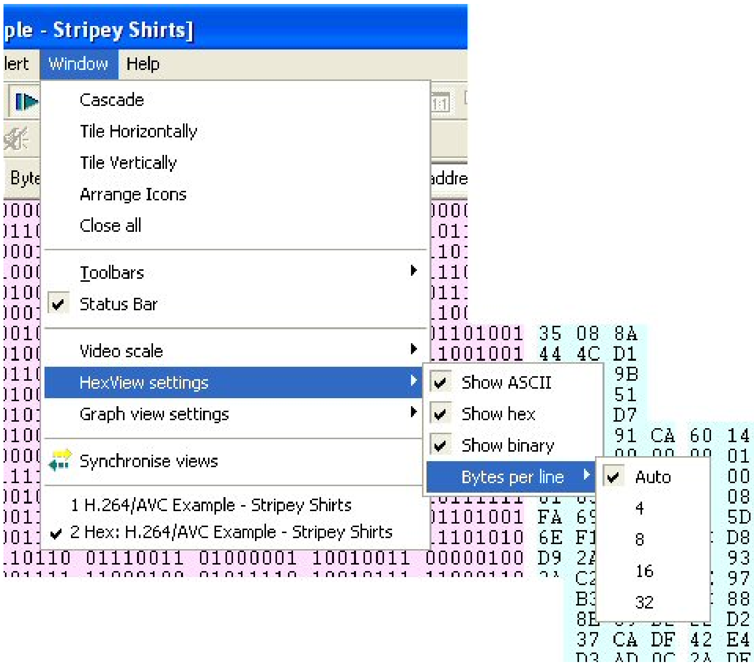
50%; 100% (1:1); 200%; 400%; 800%. These set the video displayed to the relevant size in the video window.

If the selected size means that the video is larger than the window size available, then a section of the video window will be displayed, with scroll bars:

When MTS4EA starts, it is set by default to 100% or 1:1 - that is, 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



Below the HexView selection the menu has four options:

- Show ASCII
- Show hex
- Show binary
- Bytes per line

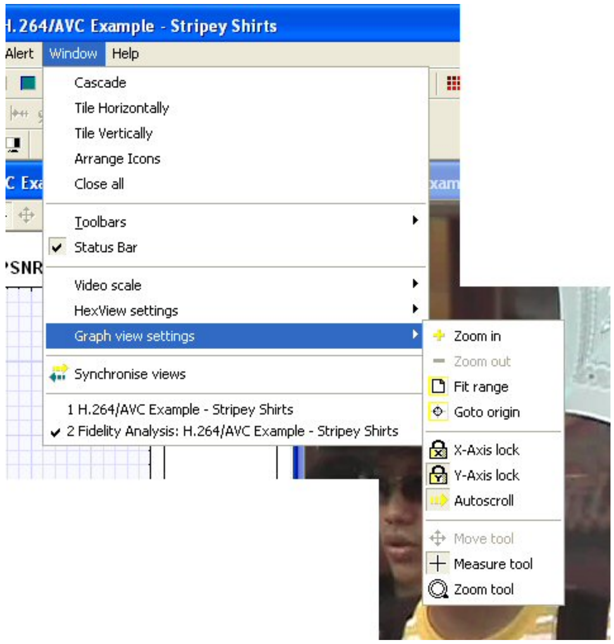
Show ASCII, Show hex, Show binary. These options individually switch on/off the display of the relevant area of the HexView.

They can be switched 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 which of the ASCII/hex/binary areas are visible
- 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 then a horizontal scroll bar is displayed along the lower edge of the HexView window

Graph view settings



Below the Graph view selection, the menu has ten options which control how the data appears on the Buffer analysis graph.

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 window
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	

Help Menu

This menu provides access to the Help information (on-line, PDF), the tutorials, configuration information, and license information (including options licensed).

Help topics

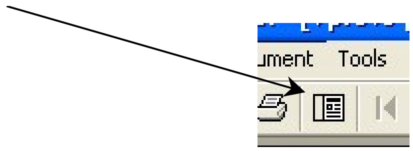
Help, organized into topics. This is the help that is displayed when the F1 key is pressed or  icon on the toolbar is clicked. Note that this Help also includes the tutorials.

User manual (PDF)

This displays a PDF version of this user manual. This is provided as:

- It is sometimes preferable to use this as a source of Help
- There is a full clickable table of contents which allows easy navigation through the whole manual
- There are thumbnails of all the 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 on the Navigation pane icon:



Then click on the Bookmarks tab to see the list of topics.

See the Adobe Acrobat documentation for details how to navigate through these views.

Tutorials (PDF)

This displays a PDF version of the tutorials on how to use/get the benefit from MTS4EA. It is strongly recommended that the user work through the tutorials. Note that 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 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 information on this.

About MTS4EA...

This displays the exact version number of MTS4EA and the copyright message, and allows access to a pop-up which displays MTS4EA build options.

Icon Toolbars

These 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 the functions to play the video, open files, etc.
- Views toolbar, containing the functions to go to the views other than the video

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

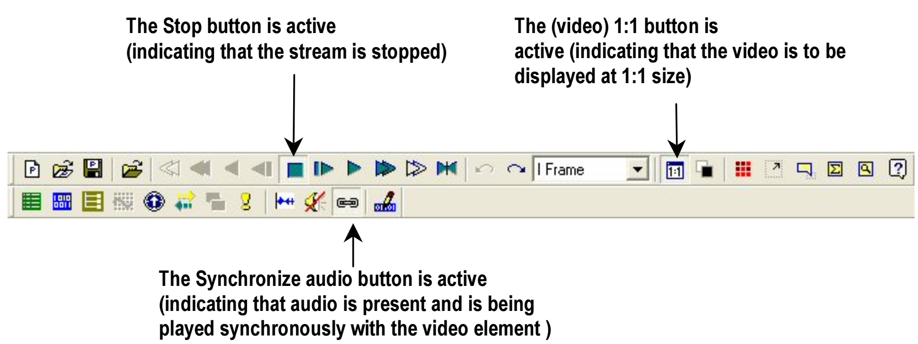
At various times, some or many of the toolbar buttons are grayed out, indicating that they are disabled, when the particular button cannot be used as the function concerned cannot be active at the time.

The only buttons available in the above example are the Open video... and Help buttons.

For example, when the HexView is opened (see *View* stream hex... `Ctrl+H` on page 6–151) the toolbar buttons concerned with the video files - such as Play, Forward, etc. - will be unavailable as these are not applicable to the HexView.

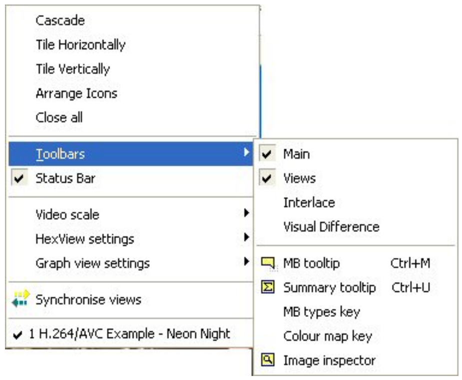
Active/inactive toolbar buttons

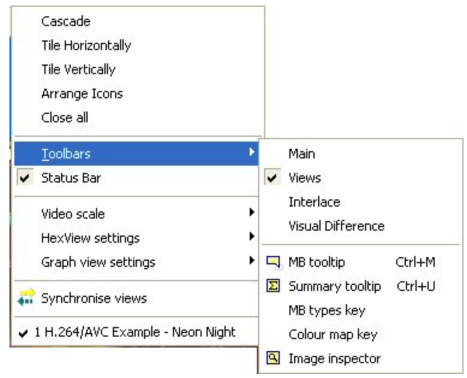
When a toolbar button is not disabled, it is active when a toolbar icon looks to be pressed in.



Displaying/hiding the Main toolbar

The Main toolbar can be displayed or hidden using the check box on the Toolbars sub-menu of the Window menu:





Toolbar displayed **Toolbar hidden**

Moving/docking

As standard, the Main toolbar is docked at the top of the window below the main menu, 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 at this point (the narrow vertical line at the left edge of the toolbar):

The Views toolbar is normally docked below this, 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 MTS4EA.

Toolbar icon	Ctrl +	Name	Description	Page
	O	Open stream...	Open a video/audio file	6–5 and 6–13
	P	Play forward	Play a video/audio file (forwards)	6–45
	F	Fast forward	Fast forward a video file	6–46
	B	Blind fast forward	Blind fast forward a video file (video blanked)	6–46
	A	Pause/Step forward	Pause a video file/advance by one frame	6–47
	S	Stop	Stop playing a video/audio file	6–47
		Pause on frame...	Pause on specific frame number	6–50
	Shift + P	Play backward	Play a video file backwards	6–48
	Shift + F	Fast backward	Fast backward a video file	6–48
	Shift + B	Blind fast backward	Blind fast backward a video file (video blanked)	6–48
	Shift + A	Pause/Step backward	Pause a video file/ step back by one frame	6–48
	K	Skip forward	Skip to next I-frame/forward n frames/forward n seconds	6–49
	Shift + K	Skip backward	Skip to previous I-frame/backward n frames/backward n seconds	6–49
	1	1:1	Set video scale: Best fit or 1:1	6–203
	W	Black/White	Set overlay digits to black/white	6–110
	Y	MB types	MacroBlock type overlay	6–86
	E	Motion vectors	Overlay motion vectors display	6–91
	M	MB tooltip	Open/close the MacroBlock tooltip	6–76
	U	Summary tooltip	Open/close the Summary tooltip	6–67
	I	Image inspector	Open/close the Image inspector	6–85
	F1	Help *	Go to Help topics	6–207

* 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.

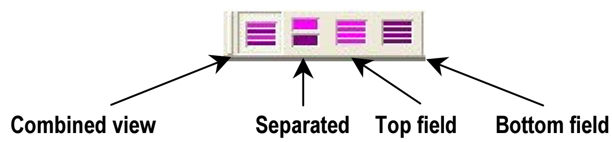
Toolbar icon	Ctrl +	Name	Description	Page
	Shift+T	View trace…	View the currently selected Trace file	6–145
	H	View stream hex…	Open the current steam in the HexView	6–151
	R	View file structure…	Open the current video/container file and view the structure in navigable tree form	6–158
		View buffer analysis…	Graphs of VBV/VCV/VMV/etc. (depends upon standard)	6–160
		Video navigator…	Show thumbnail viewer of each video frame and basic information	6–58
		Synchronize views	Synchronize all open views	6–7
		View fidelity analysis…	Show fidelity analysis	6–171
		View alert log…	Display log of alerts	6–191
		Audio waveform view…	Open the current audio stream on the Audio waveform view	6–60
		Mute audio	Turn audio off/on	6–64
		Synchronize audio	Synchronize audio and video streams	6–64
		Edit video stream…	Open the stream for editing	6–175

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 Extended Profile, Main Profile
- VC-1 Advanced, Main 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 MTS4EA (or it has been closed) it can be displayed by clicking on the Windows menu, then selecting Toolbars, Interlace.*

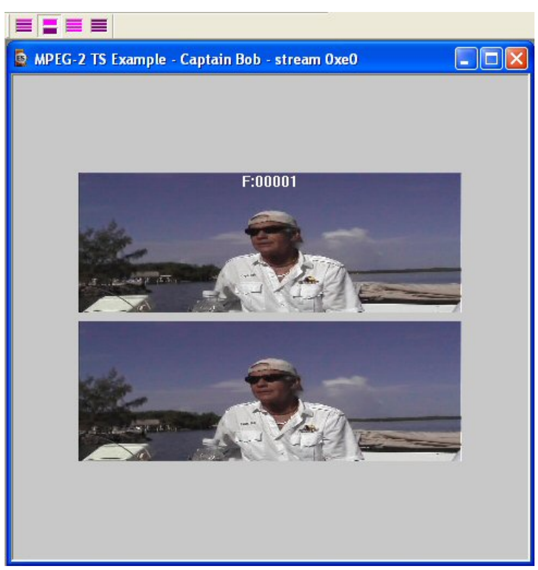
Combined frame view

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; 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.

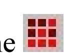
Each row is shown twice (duplicated) to produce an image to the full image height:

Bottom-field view

As above but for bottom-field.

MacroBlock types color key tooltip

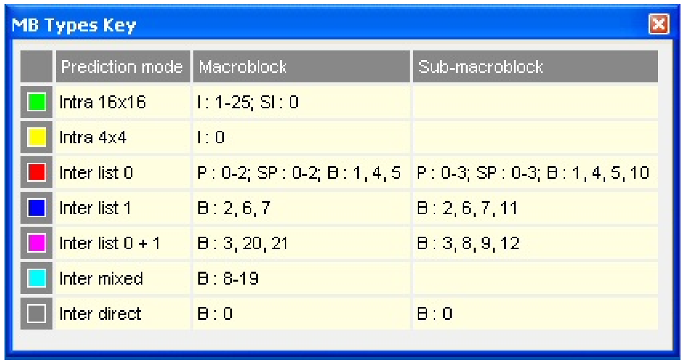
This is an information tooltip that appears when the MacroBlock Types overlay is switched on: its function is to explain the colors used.

This appears when the MB types overlay is switched 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 switched on again by going to the Window menu, selecting Toolbars, then clicking on MB types key.

This tooltip can be closed by clicking on the X at the top of the tooltip; to re-display it click the MB types overlay icon off then on.

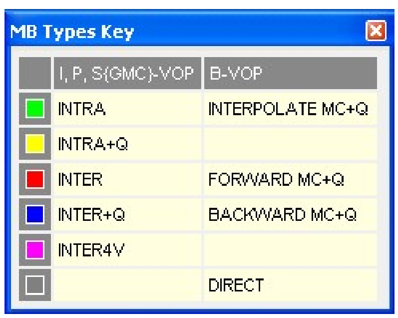


MB Types Key

	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, as 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

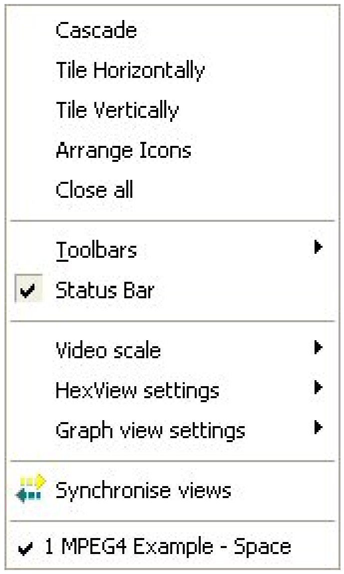
MPEG-4 example

Status Bar

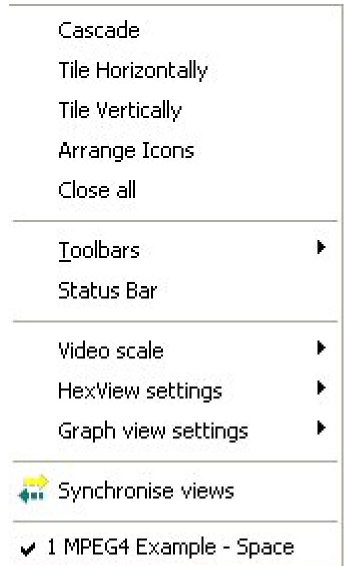
This provides:

- 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 upon the window type open and the current status of the file within that window

It is displayed hidden from within the Window menu, in the same way as the icons toolbar:

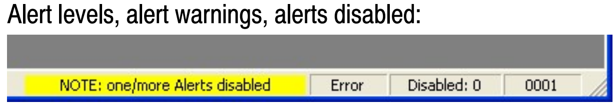
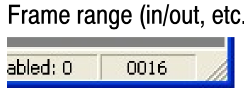
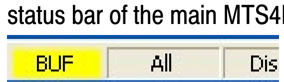


Status bar displayed



Status bar hidden

For more information on status bar indications of:

Alert levels, alert warnings, alerts disabled:  	See page 6–198
Frame range (in/out, etc.):   	See page 6–103
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 6–166 and 6–167

Ctrl Shortcut Keys

All shortcut keys are accessed by holding the Ctrl key and pressing the letter given below, except the F1 key:

Toolbar icon	Ctrl +	Name	Description	Page
	mouse	Force undock	Force undocking of tooltip (e.g. Summary/MacroBlock)	-
	1	1:1	Set video scale: Best fit or 1:1	6–203
	A	Pause/Step forward	Pause a video file/advance by one frame	6–47
	Shift + A	Pause/Step backward	Pause a video file/ step back by one frame	6–48
	B	Blind fast forward	Blind fast forward a video file (video blanked)	6–46
	Shift + B	Blind fast backward	Blind fast backward a video file (video blanked)	6–48
	C	Copy	Copy highlighted selection	6–177 and 6–184
	E	Motion vectors	Overlay motion vectors display	6–91
	F	Fast forward	Fast forward a video file	6–46
	Shift + F	Fast backward	Fast backward a video file	6–48
	G	Graph enable	Enable the graph output	6–129
	H	View stream hex…	Open the current steam in the HexView	6–151
	I	Image inspector	Open/close the Image inspector	6–85
	K	Skip forward	Skip to next I-frame/forward n frames/forward n seconds	6–49
	Shift + K	Skip backward	Skip to previous I-frame/backward n frames/backward n seconds	6–49
	M	MB tooltip	Open/close the MacroBlock tooltip	6–76
	O	Open stream…	Open a video/audio file	6–5 and 6–13
	P	Play forward	Play a video/audio file (forwards)	6–45
	Shift + P	Play backward	Play a video file backwards	6–48
	R	View file structure…	Open the current video/container file and view the structure in navigable tree form	6–158
	S	Stop	Stop playing a video/audio file	6–47

Toolbar icon	Ctrl +	Name	Description	Page
	T	Trace enable	Enable the Trace output	6–112
	Shift+ T	View trace…	View the currently selected Trace file	6–145
	U	Summary tooltip	Open/close the Summary tooltip	6–67
	V	Paste	Paste cut/copied selection in Bitstream editor view	6–177
	W	Black/White	Set overlay digits to black/white	6–110
	X	Cut	Cut highlighted selection in Bitstream editor view	6–177
	Y	MB types	MacroBlock type overlay	6–86
	Z	Undo last edit	Undo last edit in Bitstream editor view	6–177
	Tab	Switch windows	Quickly switch between open windows	6–200
	F1	Help *	Go to Help topics	6–207
	F3	Find next *	In Trace and HexView only	6–155 to 6–157
	Shift+ F3	Find previous *	In Trace and HexView only	6–155 to 6–157

* The F1, F2 & F3 keys are pressed without pressing the Ctrl key

Alt Menu Keys

All menu selections can be accessed by pressing the Alt key and the underlined letter shown on the Main menu.

For example, when the Alt key is pressed, the menu appearance changes from:

to:

NOTE. *The underline only appears when the Alt key is pressed – this behavior is determined by your Windows setting. To alter it in Windows XP, go to a blank area of the desktop, right-click, select Properties, Appearance tab, click the Effects button, then clear the check box next to* Hide underline keys until I press the Alt key...

Command Line/Batch Mode

This allows MTS4EA to be run in one of two ways:

- Command line mode
- Batch mode

Command line mode

In command line mode, MTS4EA opens with the Windows display as normal but using the files and flags specified on 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 and in this case MTS4EA opens in Windows mode, allows the user to specify a bitstream file name, then closes Windows mode and goes back to batch mode.

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 MTS4EA in command line/batch mode

To do this either:

- 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.265/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

Demo sequence	Standard	Filename
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`
H.261 stream		
Conference Room	H.261	`h261_1`
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`
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`
Microsoft® ASF files		
Beach Girl	Simple Profile	`asf_1`
Great Wall	Main Profile	`asf_2`

To use these demonstration sequences, the option `-d` is used: see *List of options* on page 6–224.

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.

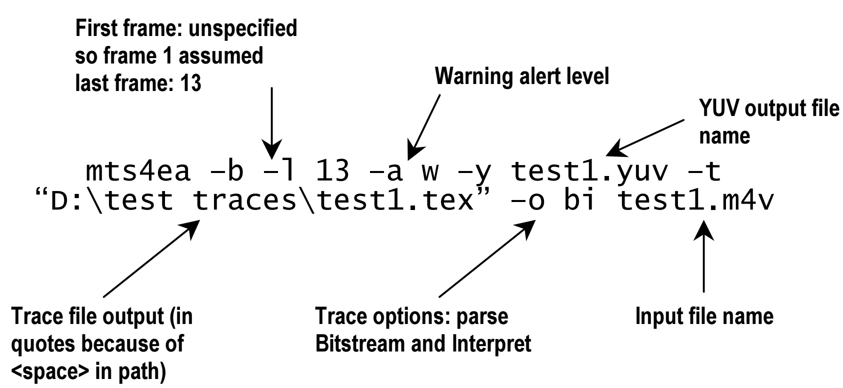
`-b`		When present, enable batch mode. If not present, command line mode is used.
`-d`		When this flag is present, it signifies that one of the demonstration tutorial video sequences as provided with MTS4EA is to be used as the input video file. The particular demonstration sequence to be used is specified by the filename as listed under **Using demonstration sequences in batch mode** on page 6–221. (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.
`-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 then 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 then 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, then there is no limit on the size of the trace file (other than the available disk space).

Option	Value (if any)	Description
-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 below are put in this file, in the order of the decoded bitstream (e.g. 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 then the trace options below are ignored.
-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.)

Option	Value (if any)	Description
-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 then 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, etc. 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 6–227).

Example command line

An example of a valid command line is:



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 from 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] in **Standards** References on page 5–20 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

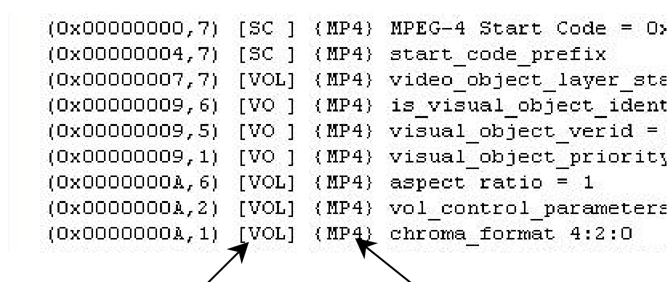
This section gives some tips and suggestions on debugging the syntax of video bitstreams syntax.

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:



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
CAB	residual_block_cabac
CAV	residual_block_cavlc
SLD	slice_data
SMP	sub_mb_pred

Three-letter code	Syntactic structure in H.264/AVC
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, $MVD_{2\text{-}4}$	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
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

Three-letter code	Syntactic structure in H.264/AVC
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**	**01**00 1000
Byte	0	1	2
Hex value	5A	3C	48

And this field has hexadecimal value 0x71 if msbf or 0x47 if lsbf.

Bitstream syntax debugging using MTS4EA

Using the various tools in MTS4EA, bitstream syntax debugging can be quick and straightforward.

How MTS4EA does syntax checking

MTS4EA checks for literally hundreds of potential errors to ensure that the bitstream under test complies with the relevant standard. 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, that 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)
- And 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 MTS4EA for syntax debugging

There are three main tools for syntax checking:

- The Alerts provided when displaying the video sequence using MTS4EA (see **Description** of Alert levels on page 6–187)
- The Trace/Parse Bitstream function (see **Parse** Bitstream on page 6–116)
- The Trace/Interpret function (see **Interpret** on page 6–119)

Essentially, of the three principal tools provided within 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** MTS4EA does syntax checking above)
- 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 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 stared with debugging the video bitstreams you have generated.

After a period of time, you will doubtless 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.

It also means that if you have to Trace out all the data of a whole video sequence into a Trace file then 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 MTS4EA, looking at the same point as MTS4EA is showing the video for or has printed out using the Trace out functions.

To use the HexView, see *View* stream hex... `Ctrl+H` on page 6–151.

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 then Parse Bitstream after, if required - it all depends upon the nature of the errors involved.

For each of these steps, follow the procedure given below:

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 carefully the message, noting the position (byte address) and bit position given. (See *Explanation of Fatal/Error/Warning/Info display* on page 6–180 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, then play the video sequence and see if any Errors or Warnings occur. If they do not, then go to Trace/Parse *Bitstream debugging* below.
6. If there are Errors and/or Warnings, then stop playing the video; switch on the Summary Tooltip (see **Summary** Tooltip Ctrl+U on page 6–67) 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 `Ctrl+T`, **Frame** range tab on page 6–113).
2. Set the Trace filename (*Filename* on page 6–115) and the maximum Trace File size (*File* size limit (available disk space) on page 6–116).
3. Set Parse Bitstream on and all other Trace functions off.

4. Put 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 given in **Parse** Bitstream on page 6–116.

7. Search for Errors and Warnings (using the Find next button in the View trace... window).

8. If there are no Errors or Warnings then 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-hand 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 actual 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 will almost certainly be the case with an Out of Sync error.

For example:

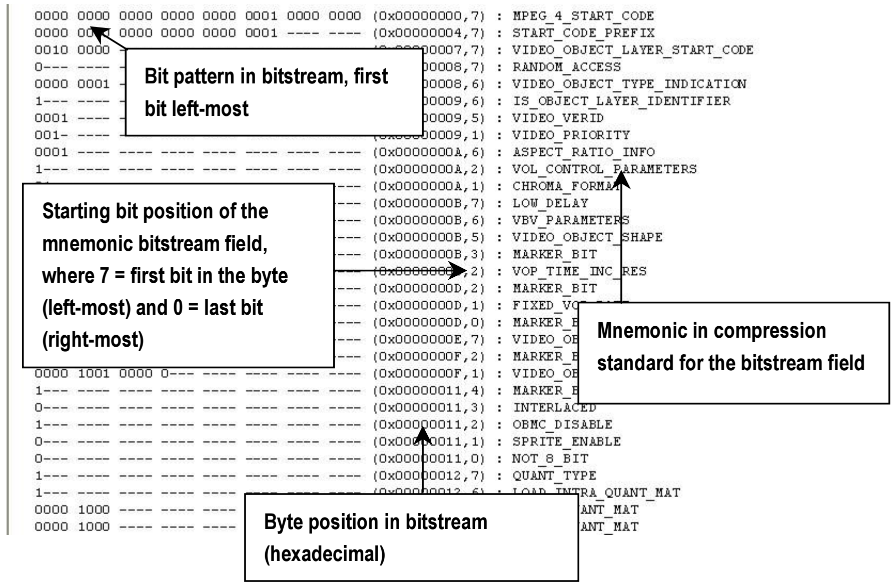
- In MPEG-4, if using video_plane_with_short_header() (MPEG-4 standard section 6.2.5.2)
- And pei = 1 then following this should be eight bits of a psupp field (these can be repeated), followed by the gob_layer(), gob_resync_marker (17 bits, 0000 0000 0000 0000 1)
- However, if pei = 0 then the gob_resync_marker 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 which 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).

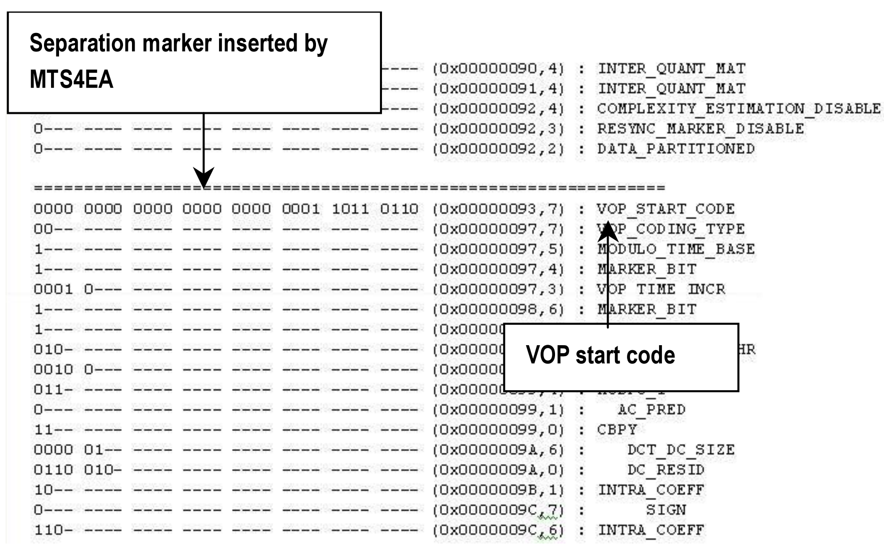
11. Check the three-letter codes in the square brackets used by MTS4EA and reported in the trace output, such as [SC], as given in **Three-letter** codes for H.264/AVC on page 7–2, and that the three-letter code corresponds with the relevant standard section.

Parse Bitstream example outputs

See also **Parse** Bitstream on page 6–116 for more information.



MPEG-4 example, at start of bitstream



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 `Ctrl+T`, **Frame** range tab on page 6–113.)
2. Set the Trace File name (Filename on page 6–115) and the maximum Trace File size (File size limit (available disk space) on page 6–116).
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 then see a Trace out file similar to that given in **Parse** Bitstream on page 6–116.
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-hand side of the trace file and relate this to the video standard you are using to see if you can understand discover the problem.

NOTE. *It is very important to understand that the actual 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 will almost certainly 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

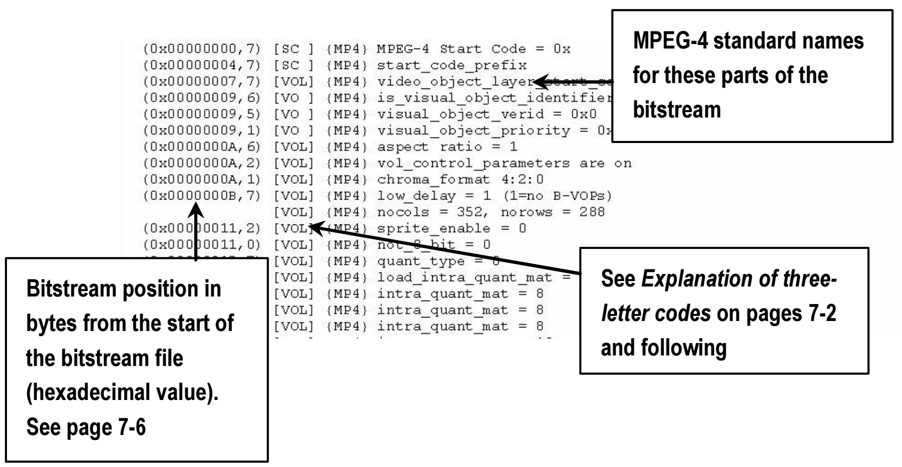
- This would appear to be an error in the gob_resync_marker (that is because the 17th bit after pei=0 would not be set to 1)

9. Check that there are no elements of a standard being used which 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]. Search for the standards specifiers used by MTS4EA.

10. 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.

Interpret example outputs

See also **Interpret** on page 6–119 for more information.



MPEG-4 Example, at start of bitstream

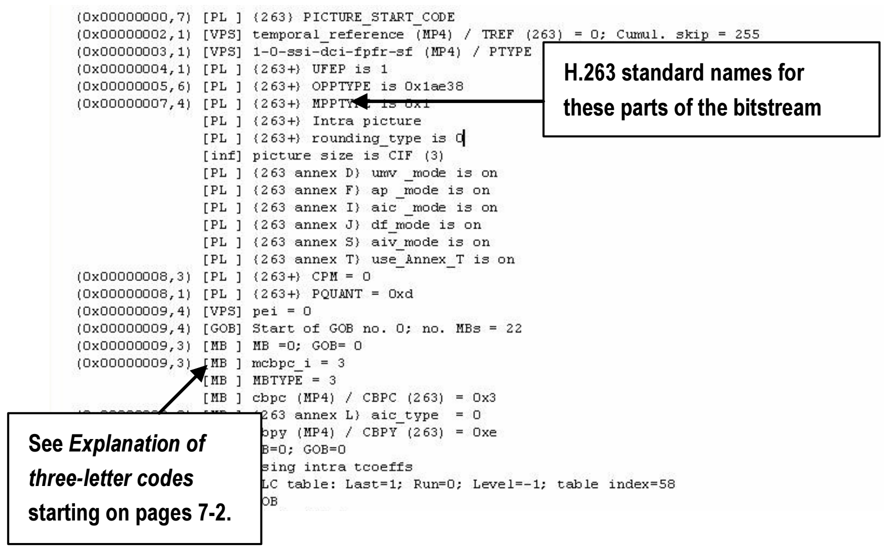
Bit position where the value starts (7=left-most; 0=right-most). See *Explanation of Bit/bit start* on page 7-6

```
(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



H.263 example at start of bitstream

Appendices

Appendix A: Decoder Plugins for MTS4EA

This section covers use of MTS4EA decoder Plugins.

Decoder Plugins for MTS4EA

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 the user may wish to do this:

- There are various divergences between the outputs of different decoders due to rounding errors or different floating-point arithmetic used, as the video standards do not completely define these. The differences given mean that the output given by MTS4EA - particularly using the YUV output in batch mode (see *Command Line/Batch Mode* on page 6–220) - 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–3 for an example.
- Sometimes the video standard has errors and differs from common implementations, and Tektronix is aware that 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 MTS4EA function within a user's own decoder, to guarantee that numerically identical results are output by MTS4EA decoder and a user's 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 MTS4EA. The DLLs are organized in their own folders.

Where a DLL is not dependent upon a particular video standard then 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 DLL supplied.

To view the documentation, open the file: `index.html` using a browser.

The API to the DLLs, file lists, structures used, data fields and globals are all accessed by clicking on the links concerned.

`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 DLL concerned. 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 which are listed in the include folder which are also needed by the included* `.h` *file will be included anyway within the* `.h` *file.*

An example of this is the `vpIdct.h` which includes `MTS4EA.h`

Format for Generating Decoder Plugins

The Decoder Plugins have been written using Microsoft Visual C 6.0.

The Plugins are all Microsoft Windows DLLs.

The only versions of Windows supported are given in *Software, Hardware and User Prerequisites* on page 3–1.

NOTE. *No other development tools or variants to these are supported for the Decoder Plugins.*

Use of Decoder Plugins

In order to use a specific Plugin copy it from the plugin folder to the folder where MTS4EA executable is (typically `C:\Program files\MTS4EA v1.3`).

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 a user to substitute their own IDCT function instead of using the one provided with MTS4EA.

By using the same IDCT function in MTS4EA and in a user's own decoder, the YUV outputs from MTS4EA should be numerically identical to the YUV outputs provided by the user's own decoder.

`qs` Plugin

The MPEG-4 standard (Reference [1] in **Standards** References on page 5–20) specifies the recommended method for calculating quarter sample accuracy predicted blocks (relating to motion vectors). However, the standard is non-optimal 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] in **Standards** References on page 5–20), 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.

MTS4EA uses the version as specified in the standard, and MTS4EA will continue to track the standard and so will change only if the standard is changed. However, the user may substitute their own quarter sample Plugin DLL, to use their own or emulate the one used by Microsoft.

`gmc` Plugin

This allows a user to substitute their own GMC (Global Motion Compensation) function instead of using the one provided with MTS4EA.

The Microsoft reference decoder software (Reference [7] in **Standards** References on page 5–20) differs from the MPEG-4 standard (Reference [1] in **Standards** References on page 5–20) 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 to 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 wish 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 MTS4EA tests of the MPEG-4 Normative bitstreams and Donated bitstreams (see References [7], [8] [9] and [10] in **Standards** References on page 5–20)

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, as per Reference [9] in **Standards** References on page 5–20
- MPEG-4 Donated bitstreams: referred to in section 4.5.8 in document N3067 dated 1999-03-18; streams dated 14/07/2000, as per Reference [10] in **Standards** References on page 5–20

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 e.g. frame 1, GOB 2, MB 15 block Y3 but these are in bitstream (MS software plays the same)
hit002.m4v	Yes	Some visible errors e.g. in frame 2, GOB4, MB 1 and in frame 3 - several MBs down left hand size, e.g. 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 e.g. 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 e.g. frame 4 halo above head in GOB 0 MBs 3,4,5 and poor MBs with hand movement e.g. 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
VOP	Effectively a frame of video (MPEG-4)

Index

Index