User Manual

Tektronix

MTS4EA
Compressed Video ES Analyzer

071-1641-00

This document supports software versions:
MTS4EA, Version 3.0

www.tektronix.com

Tektronix, Inc., P.O. Box 500, Beaverton, OR 97077

Software Warranty

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 a 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 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

Tutorials

How to Use the MTS4EA

Bitstream Syntax Debugging

Appendix A - Decoder Plugins for MTS4EA

Appendix B - Tests of MTS4EA with MPEG-4 Normative and Donated Bitstreams

Glossary

Index

Preface

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

- Section 1: Introduction
- Section 2: Intended Applications
- Section 3: Software, Hardware and User Prerequisites
- Section 4: Installation and Licensing
- Section 5: Compression Standards, Video File Types
- Section 6: Tutorials

NOTE. *Even if you read nothing else from this manual, read the TUTORIAL section. It provides the basics on how to use MTS4EA, and how to get the benefit from the software.*

- Section 7: How to use MTS4EA
- Section 8: 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 disks.

The following URLs access the Web sites for the standards organizations listed (the URLs listed were valid at the time or 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 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' and 'Hints'.

- Notes are items that you should be aware of when running MTS4EA.
- Hints are extra ideas of how to get the most out of MTS4EA: you won’t necessarily need the information provided, but it could save you some time.

Number conventions

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

For example:

- 16 [i.e. decimal] = 16 decimal;
- 0x16 [i.e. 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:

- pressing the 'F1' key; or
- from the 'Help' menu, item ‘Help topics   F1’;
- clicking the Help icon on the toolbar.

User Manual and Tutorials

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

The PDF version of this user manual can be also accessed from within MTS4EA from the Help menu, item ‘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 further copies of the manual, if you wish.

Similarly, the tutorials which form part of this manual are also supplied as a separate PDF file accessible via the Help menu option.

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

Contacting Tektronix

Phone	1-800-833-9200*
Address	Tektronix, Inc. Department or name (if known) 14200 SW Karl Braun Drive P.O. Box 500 Beaverton, OR 97077 USA
Web site	www.tektronix.com
Sales support	1-800-833-9200, select option 1*
Service support	1-800-833-9200, select option 2*
Technical support	Email: techsupport@tektronix.com 1-800-833-9200, select option 3* 6:00 a.m. -- 5:00 p.m. Pacific time

* **This phone number is toll free in North America. After office hours, please leave a voice mail message.**
Outside North America, contact a Tektronix sales office or distributor; see the Tektronix web site for a list of offices.

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 video standards H.264/AVC, MPEG-4, MPEG-2, H.263+, H.263 and H.261.

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.

The 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
- 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

Software Prerequisites

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

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

A browser such as Microsoft Internet Explorer must be available. Similarly Microsoft Excel must also be present.

MTS4EA is not supported under any other operating system.

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 (irrespective of the video standard), so the faster the better. Also, the video and Trace files can be large (hundreds of Mbytes) 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 as roughly 2-4 MBytes of data are written per video frame for the more extensive Trace functions.

For most customers this is not a problem, but if you wish 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 'ramdisk'.

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 chapter.

MTS300 MPEG Test and Monitoring

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

User Prerequisites

MTS4EA is sophisticated analysis software which presents information in detail terms 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 in section 5.

Installation and Licensing

Installation and Licensing

The instructions in this section are applicable to installation on a PC, AD951A or AD953A, unless otherwise stated in the text.

In use, the MTS4EA and the purchased options are enabled using a software key (or dongle) attached either to the parallel port or a USB port. Installation of the MTS4EA does not require the dongle to be attached.

Installation Startup

1. Ensure that the unit is powered up and stable. Close all applications.

2. Insert the MTS4EA installation CD-ROM in the disk drive.

The installation program should start automatically; a browser window will open and display setup instructions (readme.html).

If the CD-ROM fails to start and the instructions are not displayed, use the following commands:

a. Select the Start ➔ Run on the Windows taskbar.

b. In the Run dialog, enter 'd:\setup.exe' (where 'd' is the drive letter of the CD-ROM drive; use the appropriate drive letter if yours is different).

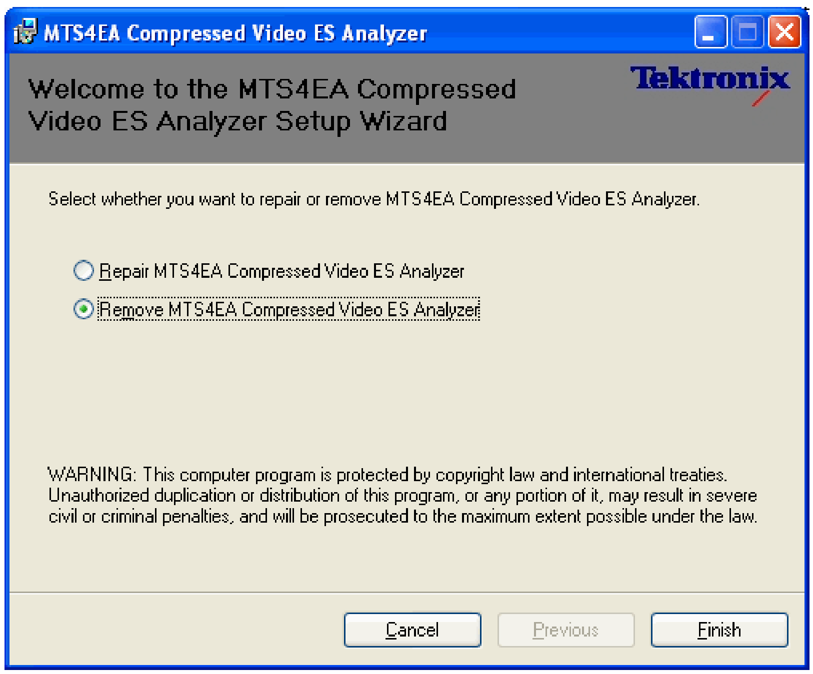
c. Select OK to run the setup program.

3. On the setup instruction page, either scroll to the bottom of the page or select the 'Install MTS4EA link at the top of the page. This causes a jump to the bottom of the page. Select 'Install MTS4EA'.

Remove Previous Versions

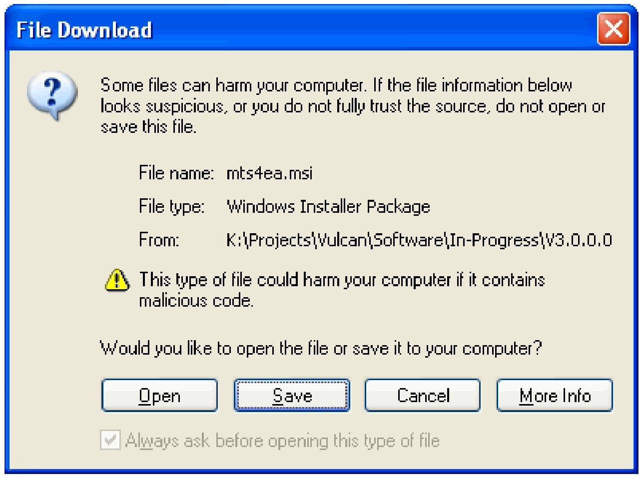
NOTE. *The previous version (or demo) of the MTS4EA must be removed before installing the current version.*

1. If a previous version (or demo) of the MTS4EA exists on the unit, the following dialog is displayed:

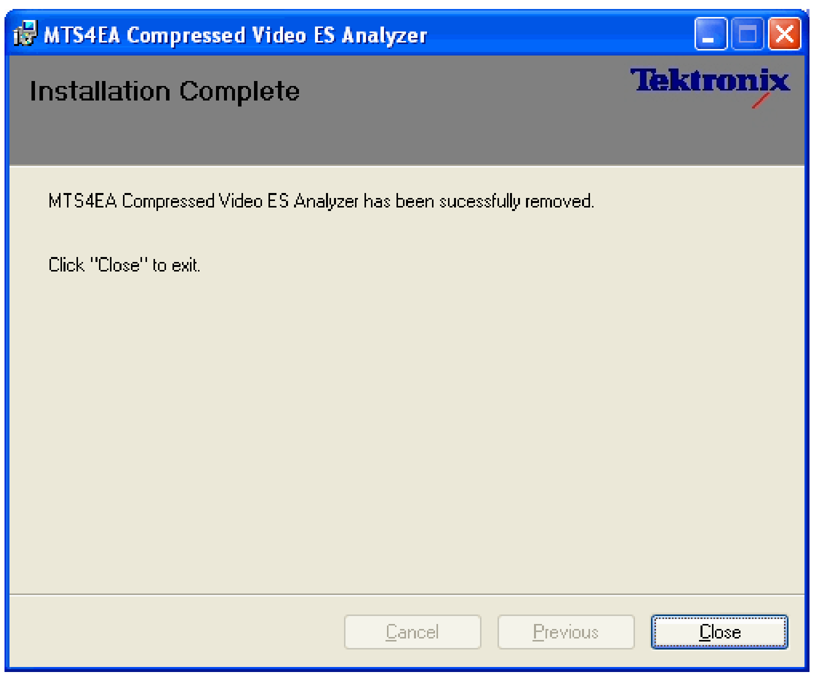


2. Select the 'Remove' option and the 'Finish' button.

3. When the file dialog is displayed, select 'Open'.



4. The following dialog is displayed when the application has been removed.



5. Select 'Close' to close the dialog.

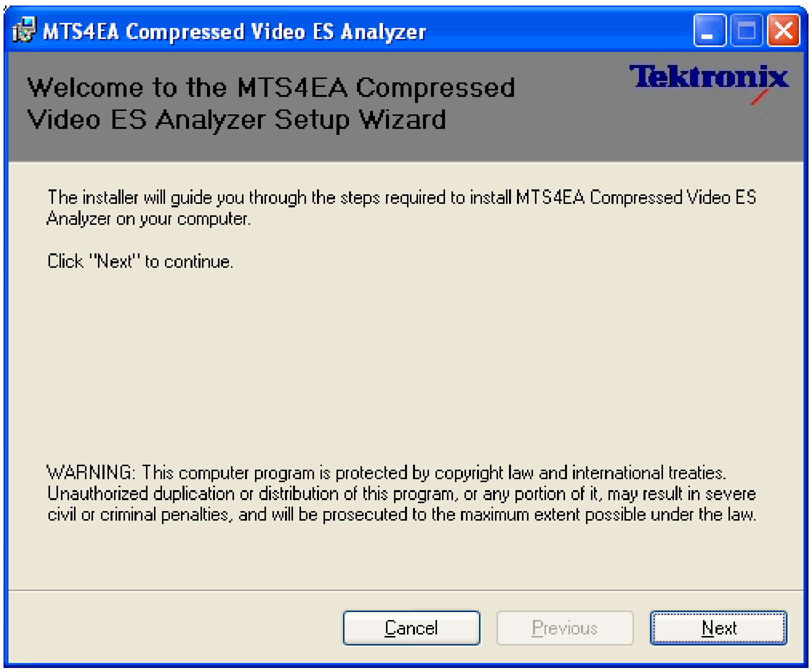
6. Close the browser containing the installation page to complete the process.

Installation

The MTS4EA installation process will install the following:

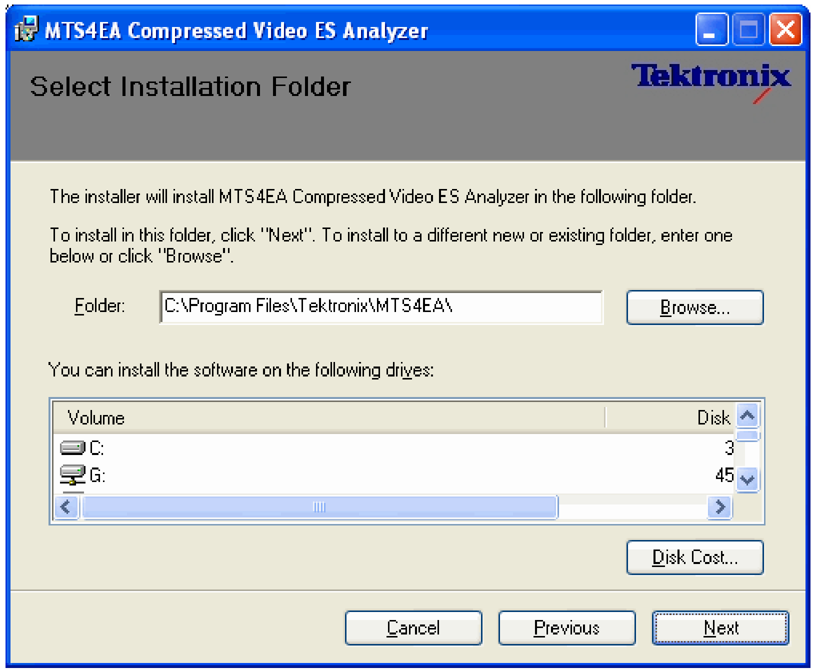
- MTS4EA application, including the purchased options
- Adobe Reader: required
- Software Key drivers (where necessary)

1. If no previous version exists on the unit, the following dialog is displayed after completing the *Installation Startup, page 4-1*:



2. Click 'Next' to continue.

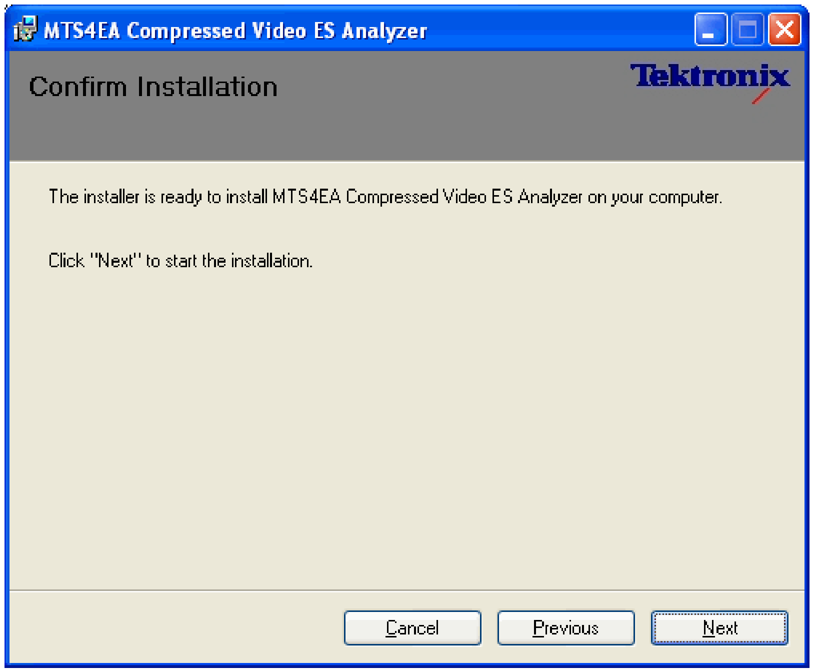
3. The 'Select Installation Folder' dialog displays the default location of the installation.



If a different location is desired, enter the path directly or browse to the required location and select.

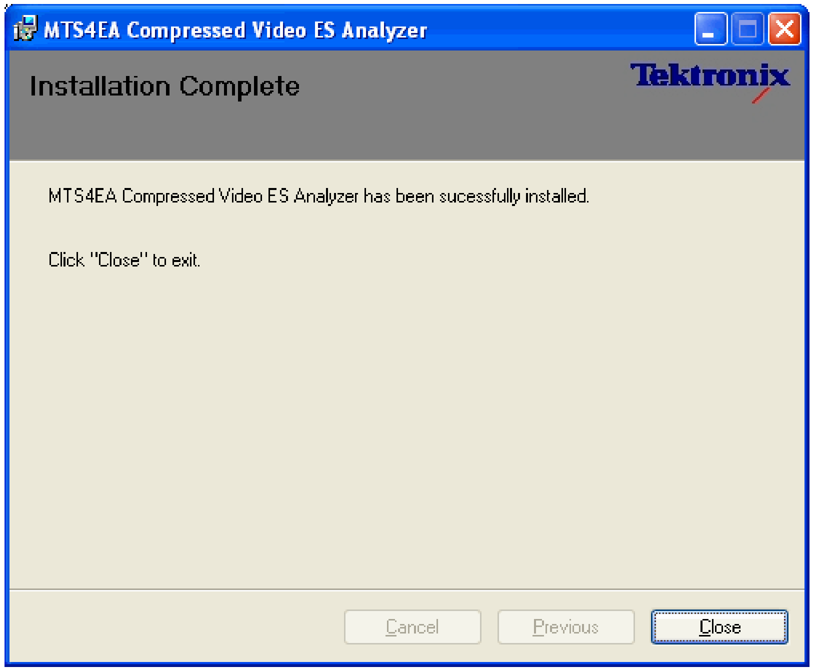
4. Click 'Next' to continue.

5. In the 'Confirm Installation' dialog, select 'Next' to start the installation.



The installation process can take up to five minutes depending on the performance of the unit.

6. When the process is complete, the 'Installation Complete' dialog is displayed.



7. Select 'Close'

8. Close the browser containing the installation page to complete the process.

Software Key

The MTS4EA software requires that a dongle (software key) is installed so that it can be accessed and the purchased options enabled.

The dongle may come in the form of a parallel port or a USB (Universal Serial Bus) device.

A dongle is a small hardware device that is connected directly to the parallel port or USB port of the host PC. Any printer compatible with the installed operating system can be connected through the dongle.

Parallel Port Dongle **USB Port Dongle**

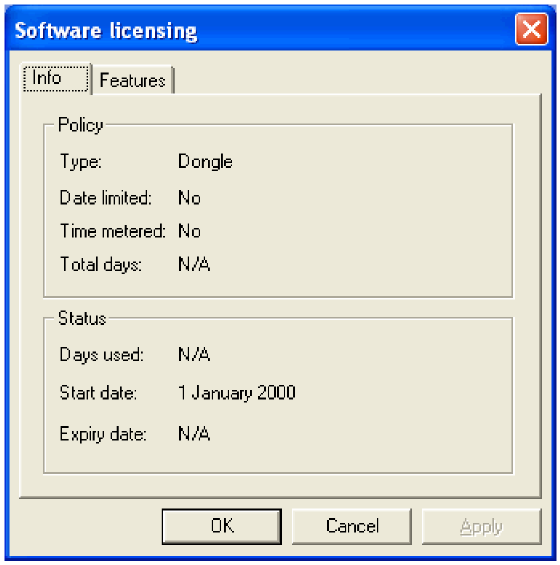
The dongle is not required to enable installation of the MTS4EA, but it must be installed before a stream can be opened.

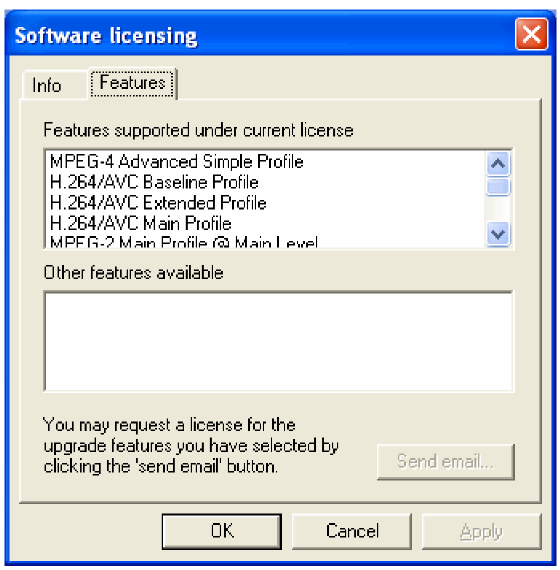
MTS4EA Options

(To be included.)

MTS4EA License Manager

The status of which options are licensed can be viewed using the 'License manager...' selection on the 'Help' menu. The Software Licensing dialog has two tabs.





Info tab

This shows the status of the license you have.

Features tab

This shows:

- which features are already licensed, and
- which other optional features are not yet licensed.

By selecting/checking the box next to the 'Other features available' that you would like to have, then clicking send e-mail you can request these features (the appropriate fee would also have to be paid first).

Release Notes

There may be last-minute information on the MTS4EA CD which gives additional information that you should consider.

If there is any such information then this will be in the file '`release_notes.txt`' in the '`\documentation`' folder on the CD.

If there are Release Notes then this file can be viewed using the '`README.HTML`' 'file (which opens automatically during installation): to see the release notes, click on ‘Documentation’ in the gray title bar at the top, and then click on the ‘View Release Notes’ link.

.

Compression Standards and Video File Types

Compression Standards and Video File Types

Supported Compression Standards

The video compression standards supported by MTS4EA are:

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

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

- MP4 (MPEG-4/H.263 Baseline only)
- 3GPP (MPEG-4/H.263 Baseline only)
- VOB/MPEG-2 Packet Stream (MPEG-2 only)

MTS4EA will also open 'YUV' format files: see *Opening a YUV video file (any file extension)*, page 7-24 for more information.

H.264/AVC

MTS4EA supports the following elements of the H.264/AVC standard (see *Standards References*, page 5-14, reference number [13]).

H.264/AVC Profiles, Levels. The following Profiles and Levels are supported, subject to the restrictions given in *MPEG-4 Tools*, page 5-5:

- 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

H.264/AVC Tools.

(See *Standards References*, page 5-14, reference number [13])

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

Tool	Baseline Profile	Extended Profile	Main Profile
Profile and level indications			
profile_idc	66	88	77
constraint_set0_flag	must be 1	0 or 1	0 or 1
constraint_set1_flag	0 or 1	0 or 1	must be 1
constraint_set2_flag	0 or 1	must be 1	0 or 1
level_idc	10 - 51	10 - 51	10 - 51
Slice types			
I	Yes	Yes	Yes
P	Yes	Yes	Yes
B	*- not allowed -*	Yes	Yes
SP	*- not allowed -*	Yes	*- not allowed -*
SI	*- not allowed -*	Yes	*- not allowed -*

NAL unit types			
1 - Coded slice of a non-IDR picture	Yes	Yes	Yes
2 - Coded slice data partition A	*- not allowed -*	Yes	*- not allowed -*
3 - Coded slice data partition B	*- not allowed -*	Yes	*- not allowed -*
4 - Coded slice data partition C	*- not allowed -*	Yes	*- not allowed -*
5 - Coded slice of an IDR picture	Yes	Yes	Yes
6 - Supplemental enhancement information	Yes	Yes	Yes
7 - Sequence parameter set	Yes	Yes	Yes
8 - Picture parameter set	Yes	Yes	Yes
9 - Access unit delimiter	Yes	Yes	Yes
10 - End of sequence	Yes	Yes	Yes
11 - End of stream	Yes	Yes	Yes
12 - Filler data	Yes	Yes	Yes
Structural			
Data partitioning (NALU type 2-4)	*- not allowed -*	Yes	*- not allowed -*
Interlace (frame_mbs_only_flag)	*- not allowed -*	Yes (L 2.1-4.1)	Yes (L 2.1-4.1)
Arbitrary slice order	Yes	Yes	*- not allowed -*
Slice groups (num_slice_groups_minus1)	1-8	1-8	1
Redundant coded pictures (redundant_pic_cnt_present_flag)	Yes	Yes	*- not allowed -*
Weighted prediction			
weighted_pred_flag	must be 0	can be > 0	can be > 0
weighted_bipred_idc	must be 0	can be > 0	can be > 0
Entropy coding (entropy_coding_mode_flag)			
CAVLC	Yes	Yes	Yes
CABAC	*- not allowed -*	*- not allowed -*	Yes
B frames			
direct_8x8_inference_flag	*- not allowed -*	1	0 (L 1-2.2) 1 (L 3-5.1)
MinLumBiPredSize	*- not allowed -*	only L 3.1-5.1	only L 3.1-5.1

MPEG-4

MTS4EA supports the following elements of the MPEG-4 standard
(See *Standards References*, page 5-14, reference number [1]).

MPEG-4 Profiles, Levels. The following Profiles and Levels are supported, subject to the restrictions given in *MPEG-4 Tools*, page 5-5:

- 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 *Standards References*, page 5-14, reference number [6])
  - Level 4
  - Level 5
- Main Profile
  - Level 2

Note that Level 0 is an addition to Simple Profile which is not in the MPEG-4 standard reference [1]. 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 standard MPEG-4 Part 2 (Visual), reference [1] Table 9-1 for description of these MPEG-4 Tools.)

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

Shaded cells indicate 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 because the MPEG-4 standard defines it as such.*

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 *Standards References*, page 5-14, reference number [14]).

MPEG-2 Profiles, Levels. The following Profiles and Levels are supported, subject to the restrictions given in *MPEG-2 Tools*, page 5-6:

- Main Profile
  - Main Level

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 - for example, PES and VOB - see* MPEG-2 higher syntactic structures*, page 5-9.*

H.263+, H.263

The following H.263 standards are supported:

- H.263 baseline standard (see *Standards References*, page 5-14, reference number [2])
- H.263+ (see *Standards References*, page 5-14, reference number [2] 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 *Standards References*, page 5-14, reference number [3])

Permitted Video File Types/Formats

MTS4EA analyzes and displays many file types, such as:

- video files containing video 'Elementary Streams';
- 'container' files, such as MP4/3GPP files and MPEG-2 Program streams, which include video, audio and other data;
- MTS4EA Trace files (which have a '`.vpt`' file extension);
- a YUV format video file
- any data file, using the 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.

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 which MTS4EA analyzes should conform to the relevant part of the MPEG-4 standard (see *Standards References*, page 5-14, reference number [11]).

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 *File Menu*, page 7-13).

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*, page 7-148).

3GPP files

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

3GPP files should conform to the standard concerned (see *Standards References*, page 5-14, reference number [12]).

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 for MP4 files. For information on opening 3GPP files, extracting and saving the video streams (see *Open video… Ctrl+O*, page 7-13).

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*, page7-148).

MPEG-2 higher syntactic structures

MPEG-2 Program Stream 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 (see *Standards References*, page 5-14, reference number [15])

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 *Standards References*, page 5-14, reference number [15]), the Program Stream file will start with a 32-bit start code: 0x000001ba; that is, 0000 0000 0000 0000 0000 0001 1011 1010.

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

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

HINT. *The user can specify the different VOBs which 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.

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.*

See *Standards References*, page 5-14, reference number [16].

NOTE. *Encrypted (scrambled) VOBs cannot be opened.*

Compressed video file format

The format 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).

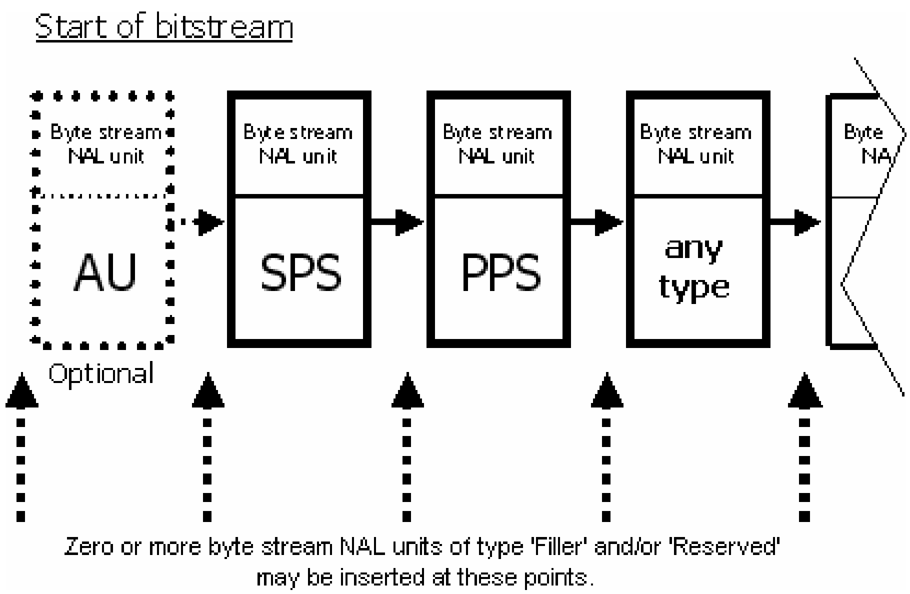
The compressed video elementary stream file cannot contain audio, although the original container file (such as MP4 or 3GPP) may have contained audio.

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 *Standards References*, page 5-14, reference number [13]), the video file must start as given in the diagram below.

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 6 bytes in an Extended profile byte stream file will be:
`00 00 00 01 67 58` (all values in hexadecimal)

MPEG-4 Elementary Stream file format. In accordance with the MPEG-4 standard (see *Standards References*, page 5-14, reference number [1]), the file will start with a valid header start code which 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 following information)

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 *Standards References*, page 5-14, reference number [1], section 6.1.2 for more information.

Note that 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, that is, 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 *Standards References*, page 5-14, reference number [15], the Elementary Stream file will start with a 32-bit start code: 0x000001b3, that is, 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 preceding 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.)

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 rearrange the data order.

Video Image Size

The maximum size of the video images that can be analyzed is the maximum 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 computer (the limit to the size is actually two million Terabytes [2^{61}], that is, two 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 video file being analyzed is cached in the MTS4EA memory in the 'Step-back buffer': if the section of video to be displayed/analyzed is 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.

Standards References

[1] MPEG-4 Part 2 (Visual): standard number ISO/IEC 14496-2:2001(E); ISO title: Information technology - Coding of audio-visual objects: Part 2: Visual 2nd Edition 2001-12-01

[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

[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)

[16] DVD Standard for Video: DVD-Video Book Part 3: Video Specifications v1.13

Tutorials

Tutorials

There are 11 tutorials:

NOTE. *You should perform the tutorials in order, regardless of the video standard of interest. (Tutorial 10 relates only to MPEG-2 and interlaced video.)*

	Tutorial	Standard	Areas covered
1	Conference Room	H.261	syntax error; compression optimisation; Graphs analysis
2	Rally (250k)	H.263 baseline	errors and error log; Trace analysis; motion vectors
3	Space	MPEG-4 Simple Profile	common errors; searching for areas for codec optimisation
4	Window Car	MPEG-4 Adv. Simple Profile	common error; HexView bitstream analysis; video navigator; synchronise views; project files
5	Packet Woman	MP4 / Simple Profile / L1(2)	extract/examine ‘container’ files; Level conformance error
6	Piccadilly Circus	MP4 / Simple Profile/ L2	searching for areas for optimisation
7	Mobile Hands	3GPP / MPEG-4 Simple Profile / L1	common error; searching for areas for optimisation
8	Canary Wharf	H.264/AVC Extended Profile / L3	syntax error in PPS; Trace analysis of syntax
9	Man Walking	MPEG-4 ASP and H.264/AVC	buffer analysis in MPEG-4 and H.264/AVC; fixing problems
10	Person Track & Grenadier Guards	MPEG-2 Main Profile / Main Level	syntax errors/warnings; MPEG-2 structure analysis; interlace
11	Man Walking & Grenadier Guards	MPEG-4, MPEG-2 and H.264	fidelity analysis (PSNR etc.); visual difference; compare MPEG-2 and H.264/AVC

HINT. *You can run multiple copies of MTS4EA to allow direct side-by-side comparisons of different video streams and encoders.*

Tutorial 1: H.261 Conference Room

This tutorial focuses on:

- Basic functions
- Alerts for syntax errors (disabling, status bar)
    - Syntax error: ‘Temporal reference’ not set
    - Conclusion: not standards compliant
- Optimization: examine MacroBlock data
- Optimization: Graph outputs

It uses the supplied video clip in the 'File' menu: ‘Example files’, 'H.261 stream', ‘Conference Room’.

NOTE: *the first three slides in the first three tutorials are roughly the same - they are repeated to provide basic information in every tutorial for someone who does not start with the H.261 tutorial.*

So, your video compression works?

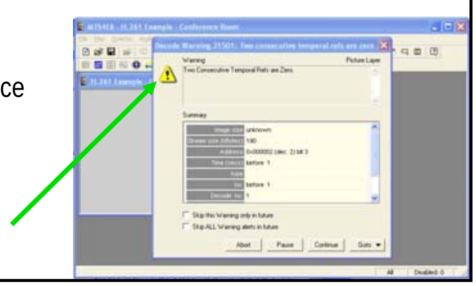
- How do you know it complies with the standards?
- If it doesn’t work well (or at all) with other vendors’ codecs, where is the problem - is it with your codec or theirs?
- Is your codec optimal?
    - Are you making the best use of available bandwidth?
    - Which frames, movement types use the most bits *and why*?
    - What changes to your codec software give the best reductions in bits used for the least reduction in visual quality?
    - What types of video content does your codec not work well with *and why*?

Basic functions

	Forwards		Backwards	
	Icon	Ctrl+	Icon	Ctrl+
Play video		P		Shift+P
Stop video		S		S
Pause/step one frame		A		Shift+A
Fast forwards/backwards		F		Shift+F
'Blind' fast forwards/backwards (no video displayed)		B		Shift+B
Pause on frame (then Fast fwd/Blind fast fwd to get to this frame quickly)				
Skip to next frame type/number/time		K		Shift+K

To begin …

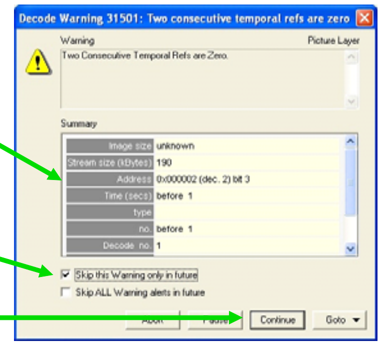
- After starting MTS4EA click anywhere to remove the splash screen (it disappears anyway after 4 seconds)
- Load the H.261 tutorial stream:
  *Note: the demo version **only** plays the example video files provided*
  - Click 'File' menu, 'Example files', 'H.261 stream', 'Conference Room'
  - The title at the top changes to 'MTS4EA - H.261 Example - Conference Room'
  - Click 'Play' menu, 'Play' or Ctrl+P, or click the toolbar icon
  - An 'Alert' pops up immediately



What is the alert?

Note: MTS4EA checks the file 'header' for errors when the file is loaded: this is why the alert pops-up before the video is played

- A 'Warning' that 'two consecutive temporal refs are zero'
  - the Temporal Reference or 'Tref' (or 'TR' in the standard) cannot be zero for two consecutive frames, but clearly it is.
- This section of the alert gives a summary of the stream up to that point
- Skip this Warning only
  (*MTS4EA* will still trigger on other warnings/errors)
- Click the 'Continue' button



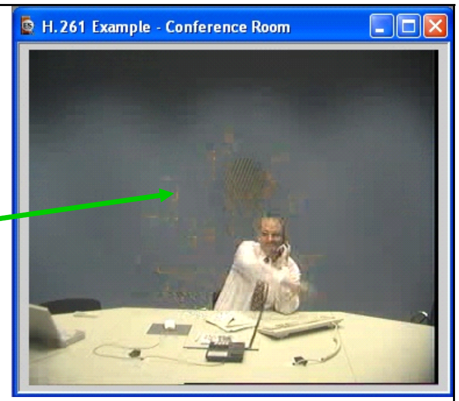
(see 'How to use MTS4EA – Alerts Menu' section of the user manual for information on Alerts)

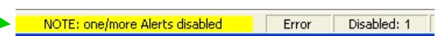
What is 'Temporal reference' in H.261?

- The standard states that the value of 'TR' is:
  = previous TR value + 1 + no. of skipped or non-reference pictures at the picture clock frequency ('PCF') (since the previously transmitted one)
- TR is 8 bits only, 0-255, at the standard PCF of 29.97 frames per second
- However, if a custom PCF is used then 10 bits are used for the temporal reference
  (the 8 LSBs are denoted as 'TR' and 2 MSBs are 'ETR' but they are taken together as a single 10-bit number to determine the temporal reference)

Continue playing the H.261 video clip

- Movement is too quick
  - you should notice that it plays too quickly (if your PC is >= 1GHz)
- Visual artifacts
  - There are some visual artifacts (unwanted noise) when the man waves his arm.
    (starting at around frame 128, continuing to frame 161, with the remains of the artifacts staying until frame 203)
- Status bar highlights that one alert is disabled



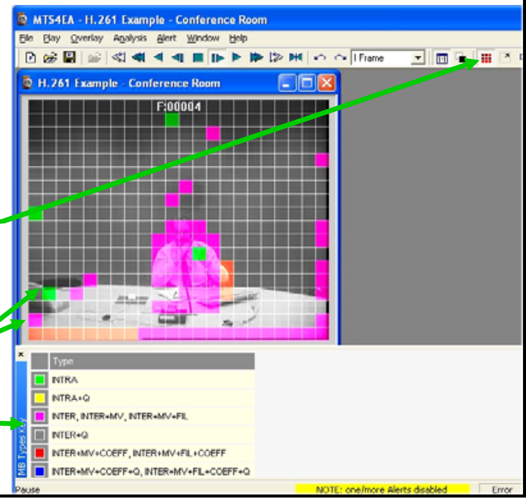


Conclusion: *not* standards compliant

- Although it will play (and probably be decoded by other H.261 decoders), the H.261 sequence has been encoded incorrectly - there are temporal issues.
- ***There are faults in the encoder: MTS4EA hasn’t fixed the problem, but it has told you what the problem is.***

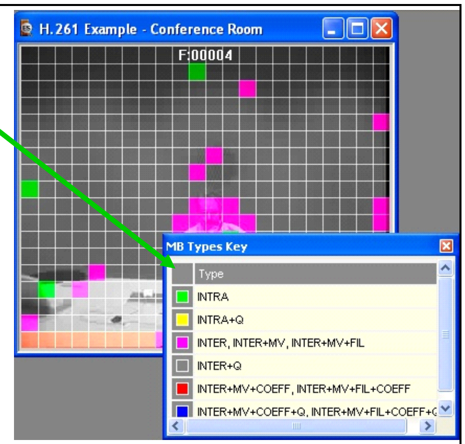
...and how good is the compression?

- It doesn't look good to the eye - are there reasons for this?
- To find out, switch on the view of the MacroBlock (MB) types
  - Click or 'Overlay' menu - 'MB Types'
  - The MBs types are color coded
  - The MB types key appears



How good is the compression? (2)

- The MB types key can be
  - Undocked (*to force undocking, hold the <Ctrl> key while dragging with the mouse*)
  - Resized
  - Switched off altogether by clicking the 'X' (top right) (to display it again, click the MB types icon twice)



There are 10 MacroBlock types in the H.261 standard: see 'How to use MTS4EA' section of the user manual for details of the MB types in H.264/AVC, MPEG-4, MPEG-2 and H.263

Look at MacroBlock data

- Now switch on the MacroBlock tooltip: click

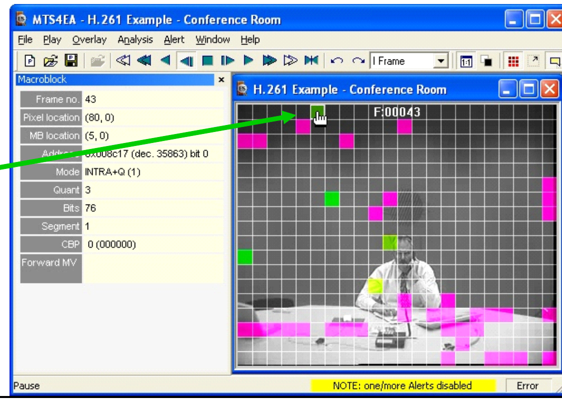
  Note: <u>Must</u> be in 'Pause' or 'Play' mode (see Status bar, bottom left)

  - MB tooltip typically 'docks' to the left edge of the MTS4EA window:

    to force undocking hold the <Ctrl> key while dragging the Tooltip with the mouse

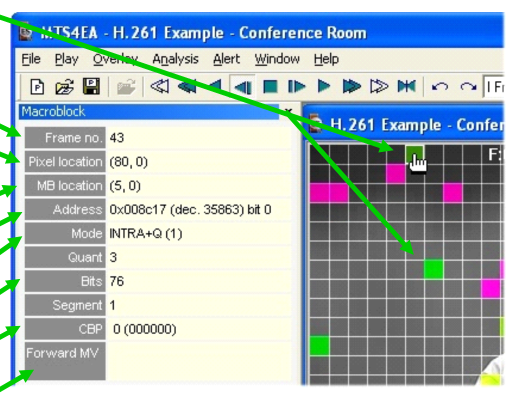
  - Moving the mouse over the video displays a white box around the MB from which data is being read

    (Look in the manual for a full description of the MB tooltip)



Look at MacroBlock data (2)

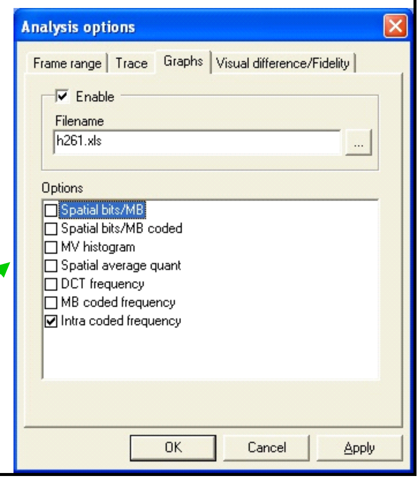
- Advance through the video, frame-by-frame
  - You will see many green Intra MacroBlocks (MBs) in the background wall - these block types use most bits
  - Frame number
  - Pixel location of top-left corner of MB
  - MB screen location (X, Y)
  - Bitstream address of MB
  - MB type/mode
  - Bits used in the MB
  - MB Coded Block Pattern
  - Motion Vectors used, if any



(See 'How to use MTS4EA' section of the user manual for information provided by the MB tooltip)

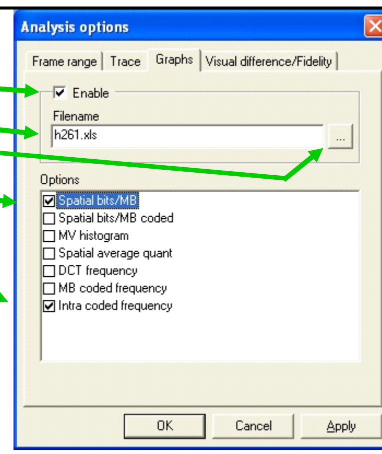
Look at MacroBlock data (3)

- It appears that many data bits are being used on the static wall in the background - check if this is the case
- Stop the video sequence (click)
- Select the 'Analysis' menu, 'Graph enable' or press Ctrl+G: the dialog box appears



Now do some graphical analysis

- Click:
  - 'Enable' (this is the default)
  - Select the output filename (select the folder with this)
  - Click 'Spatial bits/MB' and 'Intra coded frequency'
- Click 'OK', then play the video: after play ends
  - The data is exported to the '.xls' file and MS Excel is opened using this file
  - In Excel, click 'Enable macros'
  - The screen will flash as the macros run
  - At the end, there will be a tab 'Splash' with MTS4EA at the top left

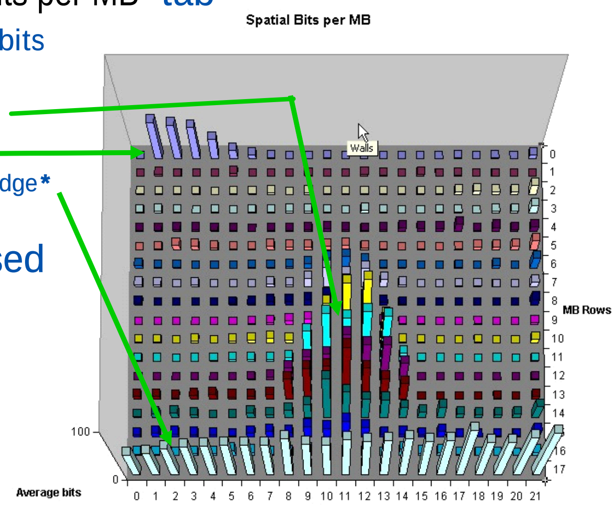


Graphical analysis (2)

➢ Click the 'Spatial Bits per MB' tab

- you can see many bits are used at
  - center, by the man
  - top left*
  - along the bottom edge*

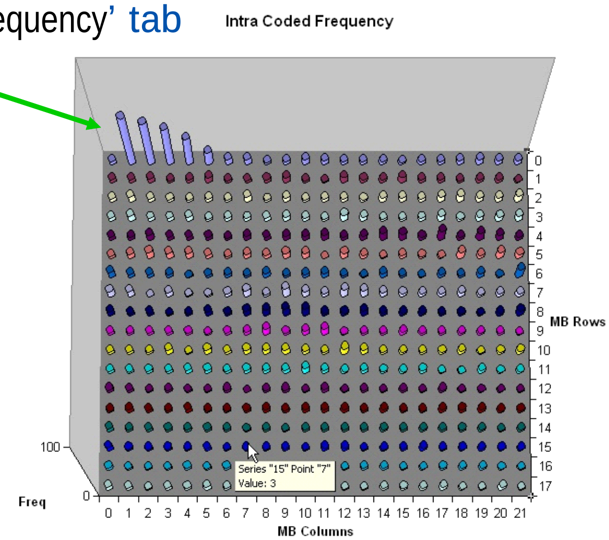
* Too many bits used in these MBs - ***clearly*** errors



Graphical analysis (3)

➢ Click 'Intra Coded Frequency' tab

- Many Intra-coded MBs at top left
- ***Clearly an error with Intra-coding in the encoder***



Summary of H.261 video clip

- Clearly there are multiple problems:
  - Standards non-compliance
    - Frequency of playing; Temporal Reference error
  - Coding inefficiency
    - See how many bits are used at the top left of the frame and at the bottom edge - investigate these areas of coding
    - Too many Intra-coded blocks - look at decision when to Intra-code

- ***MTS4EA hasn't fixed all the problems but MTS4EA has shown where to look to fix the H.261 encoder***

Tutorial 2: H.263 Rally (250k)

This tutorial focuses on:

- Basic functions
- Alerts for syntax errors
- ‘Consequential’ syntax errors/error alerts
- Trace outputs:
  - Parse bitstream
  - Interpret
  - Set frame range
- Motion vector overlay

It uses the supplied video clip in the 'File' menu: ‘Example files’, 'H.263 streams', ‘Rally (250k)’.

Note*: the first three slides in the first three tutorials are roughly the same - they are repeated to provide basic information in every tutorial for someone who does not start with the H.261 tutorial.*

So, your video compression works?

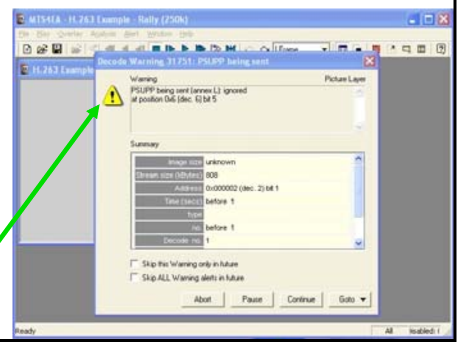
- How do you know it complies with the standards?
- If it doesn't work well (or at all) with other vendors' codecs, where is the problem - is it with your codec or theirs?
- Is your codec optimal?
  - Are you making the best use of available bandwidth?
  - Which frames, movement types use the most bits *and why*?
  - What changes to your codec software give the best reductions in bits used for the least reduction in visual quality?
  - What types of video content does your codec not work well with *and why*?

Basic functions

	Forwards		Backwards	
	Icon	Ctrl+	Icon	Ctrl+
Play video	▶	P	◀	Shift+P
Stop video	■	S	■	S
Pause/step one frame	▮▶	A	◀▮	Shift+A
Fast forwards/backwards	▶▶	F	◀◀	Shift+F
'Blind' fast forwards/backwards (no video displayed)	▷▷	B	◁◁	Shift+B
Pause on frame (then Fast fwd/Blind fast fwd to get to this frame quickly)	▶◀		▶◀	
Skip to next frame type/number/time	↷	K	↶	Shift+K

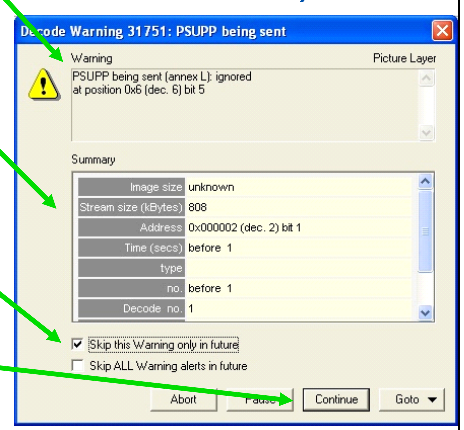
To begin…

- After starting MTS4EA click anywhere to remove the splash screen (it disappears anyway after 4 seconds)
- Load the H.263 tutorial stream
  Note: the demo version only plays the example video files provided
  - Click 'File' menu, 'Example files…', 'H.263 streams', 'Rally (250k)'
  - The title at the top changes to 'MTS4EA - H.263 Example - Rally (250k)'
  - Click 'Play' menu, 'Play' or Ctrl+P, or click the toolbar icon
  - An 'Alert' pops up immediately



What is the alert?

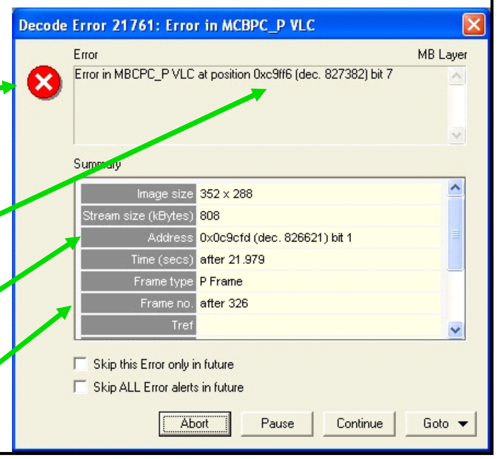
- A 'Warning' that the 'PSUPP' field is being sent but it is being ignored (part of H.263+ Annex L)
- This section of the alert gives a summary of the stream up to that point
- Skip this Warning only (MTS4EA will still trigger on other warnings / errors)
- Click the 'Continue' button



(see 'How to use MTS4EA' section of the user manual for information on Alerts)

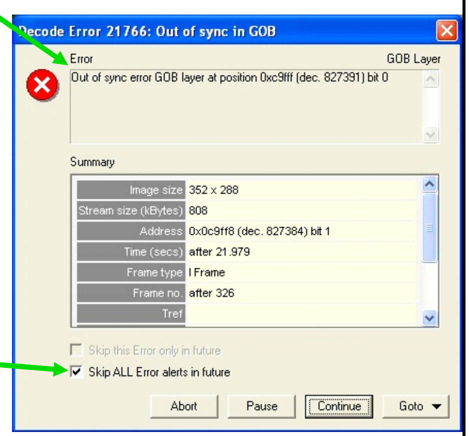
Play the video to the end…an 'Error'

- It looks ~OK, although maybe the quality is not that high?
- ...An alert in the last frame - this time an 'Error'
  - This seems a more serious problem - a syntax error in the bitstream
  - ...At bitstream byte address 0xc9ff6, bit 7
  - ...At address 0xc9cfd after frame 326 is



A 'consequential' alert

- Another alert occurs due to the 'MBCPC' error - there is an 'Out of sync' alert
- *This happens <u>very</u> often when there are syntax errors - one syntax error triggers a number of subsequent alerts (and a <u>further</u> alert follows)*
- Click 'Skip ALL…' to the go to end of the stream



Error Alert log

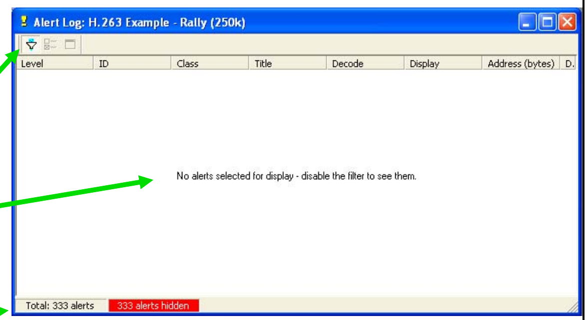
- All errors are logged in the Alert log, and can be seen by clicking on the Alert log icon:

Because previously we had clicked 'Skip ALL errors…', the Alert log view is filtered:

- Indicated by the 'filter' button being pushed in

- The message

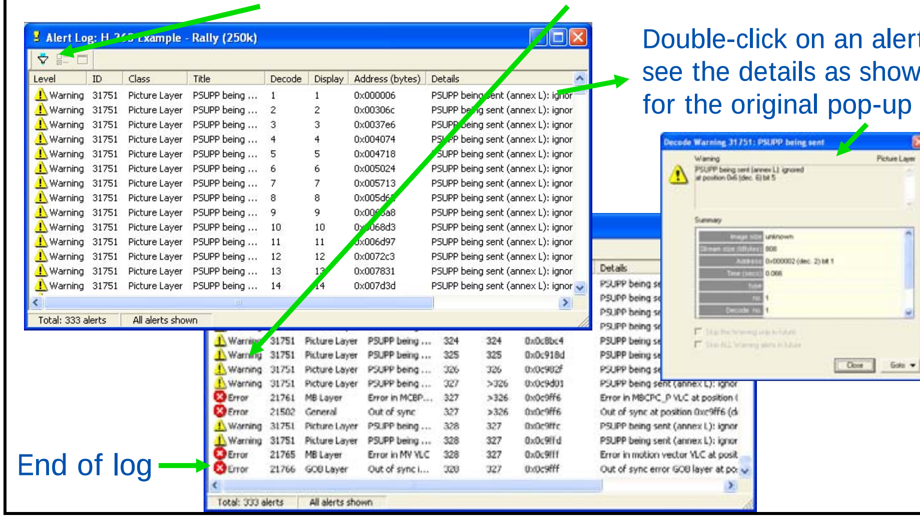
- The status bar indicates that of the total 333 alerts, all of them are hidden



Alert log (2)

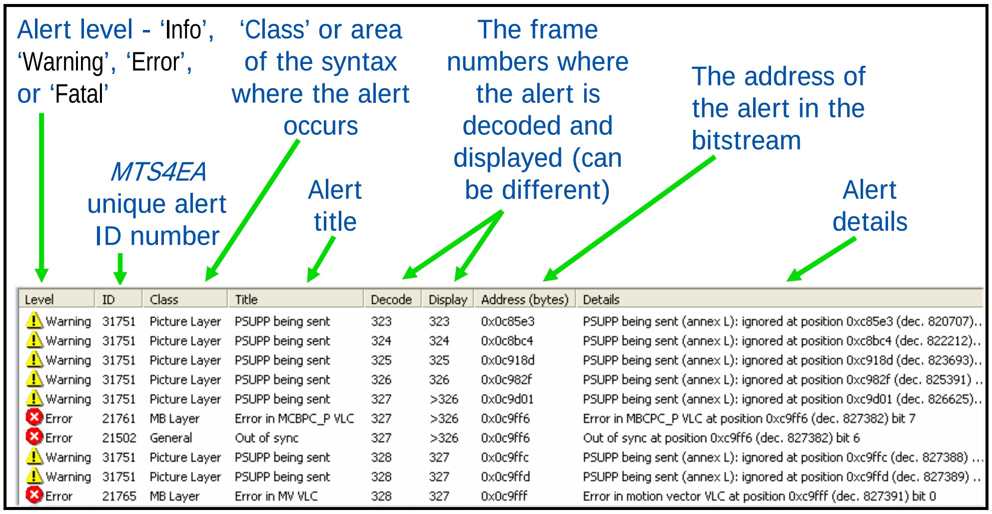
- Click the filter icon to see all Alerts:

Double-click on an alert to see the details as shown for the original pop-up



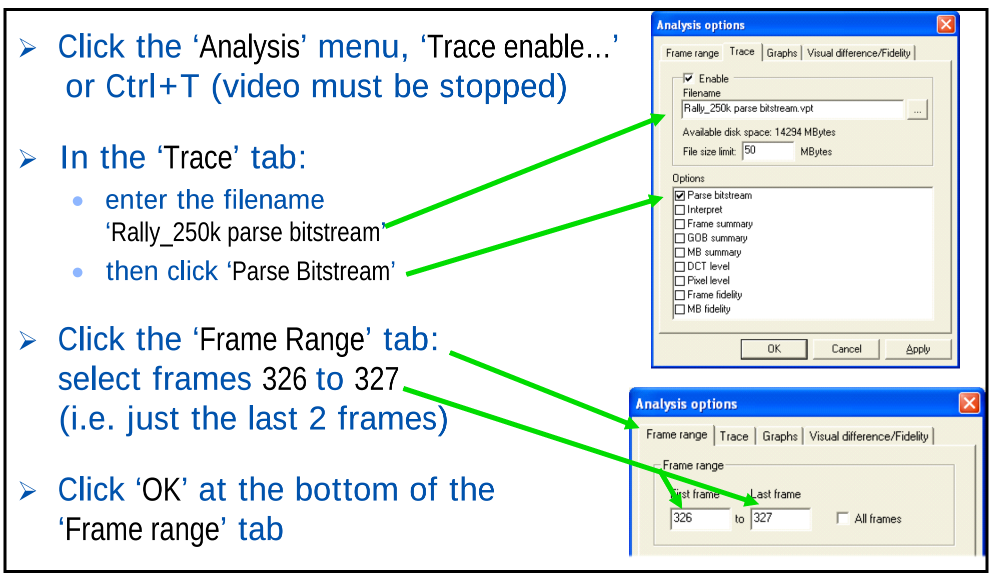
End of log

Alert log - information provided

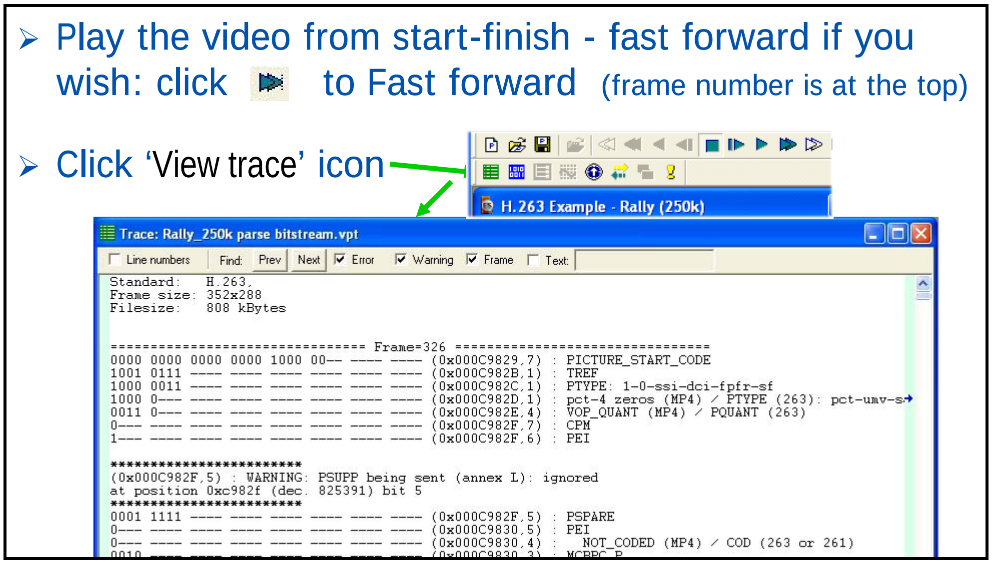


Level	ID	Class	Title	Decode	Display	Address (bytes)	Details
Warning	31751	Picture Layer	PSUPP being sent	323	323	0x0c85e3	PSUPP being sent (annex L): ignored at position 0xc85e3 (dec. 820707)..
Warning	31751	Picture Layer	PSUPP being sent	324	324	0x0c8bc4	PSUPP being sent (annex L): ignored at position 0xc8bc4 (dec. 822212)..
Warning	31751	Picture Layer	PSUPP being sent	325	325	0x0c918d	PSUPP being sent (annex L): ignored at position 0xc918d (dec. 823693)..
Warning	31751	Picture Layer	PSUPP being sent	326	326	0x0c982f	PSUPP being sent (annex L): ignored at position 0xc982f (dec. 825391) ..
Warning	31751	Picture Layer	PSUPP being sent	327	>326	0x0c9d01	PSUPP being sent (annex L): ignored at position 0xc9d01 (dec. 826625)..
Error	21761	MB Layer	Error in MCBPC_P VLC	327	>326	0x0c9ff6	Error in MBCPC_P VLC at position 0xc9ff6 (dec. 827382) bit 7
Error	21502	General	Out of sync	327	>326	0x0c9ff6	Out of sync at position 0xc9ff6 (dec. 827382) bit 6
Warning	31751	Picture Layer	PSUPP being sent	328	327	0x0c9ffc	PSUPP being sent (annex L): ignored at position 0xc9ffc (dec. 827388) ...
Warning	31751	Picture Layer	PSUPP being sent	328	327	0x0c9ffd	PSUPP being sent (annex L): ignored at position 0xc9ffd (dec. 827389) ..
Error	21765	MB Layer	Error in MV VLC	328	327	0x0c9fff	Error in motion vector VLC at position 0xc9fff (dec. 827391) bit 0

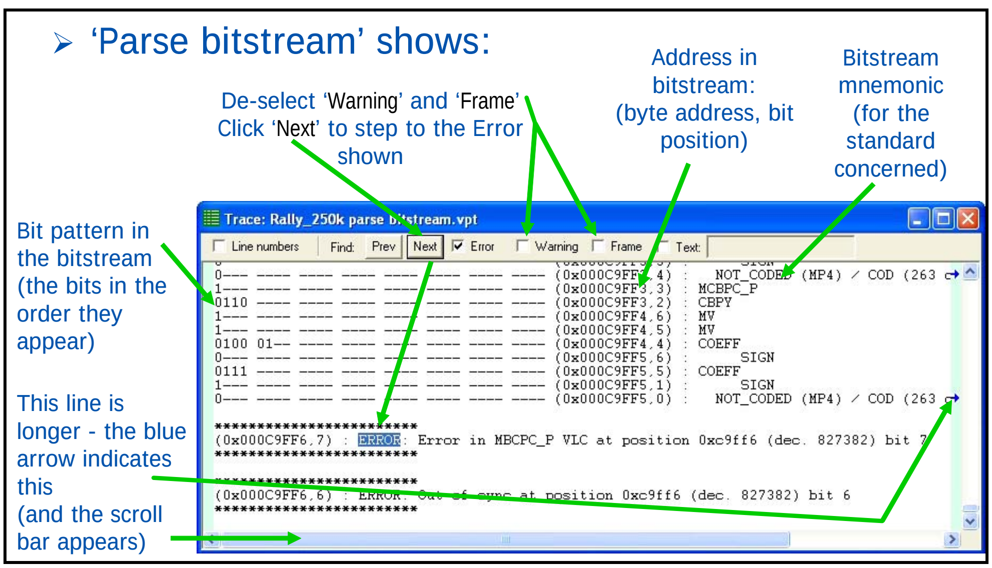
Finding out about the errors - Trace



Trace - parse bitstream



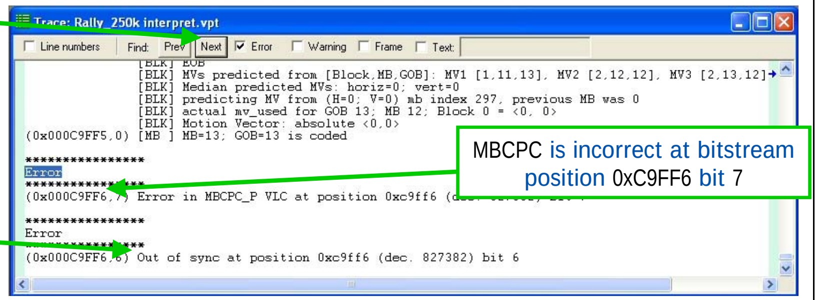
Trace - parse bitstream (2)



Trace - interpret

- Click the stop icon ; go to the Trace-enable tab; select a new filename (e.g. 'Rally_250k interpret'); step through the first 4 frames and click the View trace icon
- The Trace - Interpret shows the interpretation of the bit patterns into numbers/meanings:

Click 'Next' to step to the Error shown (or press F3)



The 'Out of sync' error follows on from the MBCPC error

Trace files result

- The Trace files have shown that the bitstream has an incorrect value for MBCPC - in fact, it appears that a Start Code[1] has been put into the bitstream by the encoder, some 9 bytes before the end of the file
- Note that the Trace files can be very large if Trace is active over many frames - that is why it is suggested always to use a range of frames

[1] actually called a 'Picture Start Code' or 'PSC' in the H.263 standard

Conclusion: *not* standards compliant

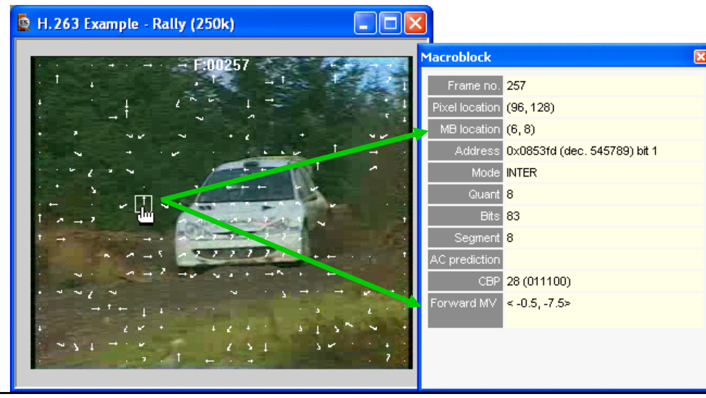
- There are two errors:
  - The warning concerning PSUPP - this is not a serious error
  - The VLC error may not to be a serious problem because it only occurs right at the end, but clearly the bitstream has not been correctly written by the encoder and this error could occur earlier in a bitstream, causing issues of interoperability with other decoders/equipment
- ***There are faults in the encoder: MTS4EA hasn't fixed the problems, but it has told you exactly where to look***

Look at the Motion Vectors

- Click in the toolbar to switch on the view of Motion Vectors ('MV') and run the video again
  - The white dots are at the center of each MacroBlock
  - The white lines indicate the Motion Vectors:
    - The MV has an arrow head at its end and is pointing to the position in the previous frame where the data for the current MacroBlock came from (or block, if 4 MVs have been used)
  - The overlay can be black instead of white - click

Look at the Motion Vectors (2)

- Now switch on the MB tooltip - click or Ctrl+M
  - Use the cursor to drag the white square to the MB being inspected - the MB shown is at (X,Y) location 6,8 and the Motion Vector values are <-0.5,-7.5> (to half-pixel accuracy)



Look at the Motion Vectors (3)

- Run through the video again; using 'File' menu, 'Pause/Step forward' (or click or press Ctrl+A)
- It looks OK
  - the Motion Vectors look fine - the direction is understandable and seems correct
  - the Motion Vector values seem fine
- Provisional conclusion: the Motion Vector implementation seems OK

Summary of H.263 'Rally (250k)'

- There are two standards-compliance issues
  - An error at the end of the sequence - not ended correctly
  - A warning at the start - not serious.
- The Motion Vectors look OK
  (Clearly more work can be done looking at optimization - see Tutorial 3 – MPEG-4 Space)
- ***MTS4EA has shown where there is a standards-compliance issue in the H.263 bitstream, but that other aspects - the Motion Vectors - are implemented OK***

Tutorial 3: MPEG-4 Space

This tutorial focuses on:

- Basic functions
- Alerts for syntax errors
- Summary tooltip
- MacroBlock overlay MB types
- MacroBlock tooltip
- Optimization
  - Frequency of Intra-coding
  - Set a frame range

It uses the supplied video clip in the 'File' menu: ‘Example files’, 'MPEG-4 Elementary Streams', ‘Space’.

NOTE. *The first three slides in the first three tutorials are roughly the same - they are repeated to provide basic information in every tutorial for someone who does not start with the H.261 tutorial.*

So, your video compression works?

- How do you know it complies with the standards?
- If it doesn't work well (or at all) with other vendors' codecs, where is the problem - is it with your codec or theirs?
- Is your codec optimal?
  - Are you making the best use of available bandwidth?
  - Which frames, movement types use the most bits *and why*?
  - What changes to your codec software give the best reductions in bits used for the least reduction in visual quality?
  - What types of video content does your codec not work well with *and why*?

Basic functions

	Forwards		Backwards	
	Icon	Ctrl+	Icon	Ctrl+
Play video		P		Shift+P
Stop video		S		S
Pause/step one frame		A		Shift+A
Fast forwards/backwards		F		Shift+F
'Blind' fast forwards/backwards (no video displayed)		B		Shift+B
Pause on frame (then Fast fwd/Blind fast fwd to get to this frame quickly)				
Skip to next frame type/number/time		K		Shift+K

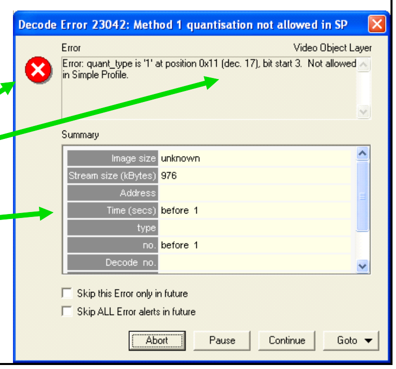
To begin…

- After starting MTS4EA click anywhere to remove the splash screen (it disappears anyway after 4 seconds)

- Load the MPEG-4 'Space' file to test:

  Note: the demo version only plays the example video files provided

  - CLICK 'File' menu , 'Example files', 'MPEG-4 Elementary Streams', 'Space'
  - *An error pop-up occurs immediately*
  - ...At bitstream byte address 0x11, bit 3
  - ...Before VOP 1 (i.e. in the header)

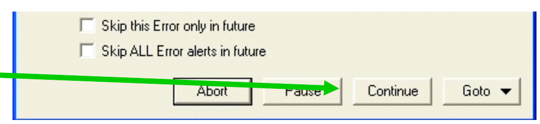


What is the error?

Note: The error pops up immediately because MTS4EA does an initial check of the file header when it loads the file - and there is an error in the header.

- The error says that method 1 quant is being used - but this stream is Simple Profile, so this is not allowed.

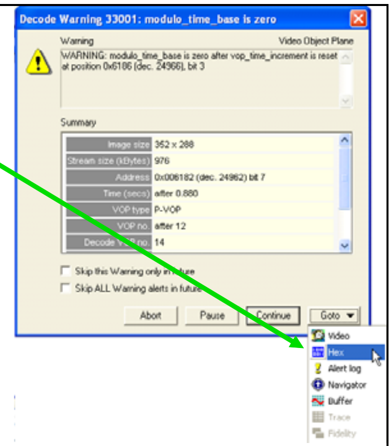
- Click 'Continue' (i.e. continue anyway)

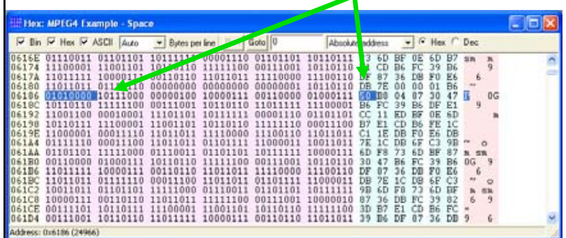


  - MTS4EA then loads the stream.
  - The title at the top changes to 'MTS4EA - MPEG4 Example - Space'
  - Click 'Play' then 'Play' or Ctrl+P, or click the icon.
  - The same error will pop-up again - this time, when the stream starts playing.
  - Click 'Continue' again on the alert pop-up.

A second alert pops up - look at Hex

- This time a warning after frame 12: look at the 'hex' of the bitstream by clicking 'Goto', 'Hex'.
- The Hex view is opened, with the error location highlighted.

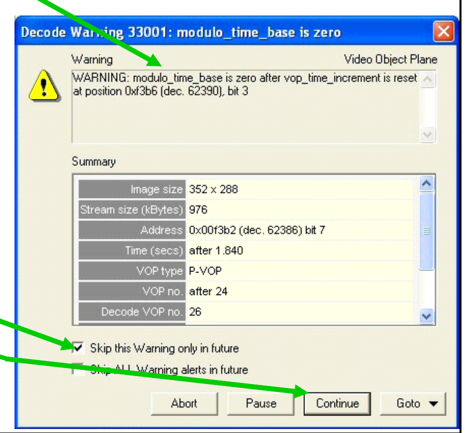




See Tutorial 4 - MPEG-4 ASP Window Car for details on how to use the HexView

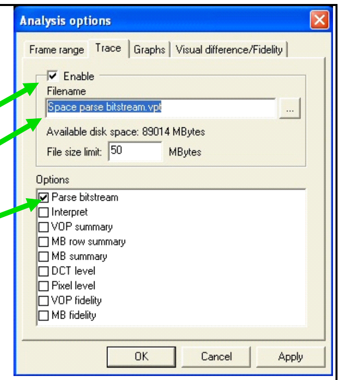
What is the modulo_time_base alert?

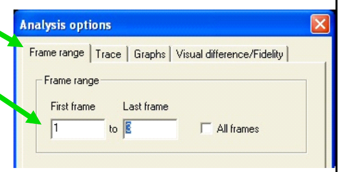
- 'modulo_time_base' error:
  - It should be set to '1' after a whole second has elapsed since the 'modulo_time_base' was last '1', at which time 'vop_time_increment' should be reset (although not necessarily to zero).
- Continue to play the video:
  - The same error occurs again in a later frame.
  - Click 'Skip this Warning…'
  - Then click 'Continue'
  - (and play to the end)
- *The video looks OK?*



Finding out about the errors - Trace

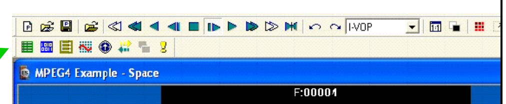
- Click the 'Analysis' menu, 'Trace enable…' or Ctrl+T (*video must be stopped*).
- In the 'Trace' tab click:
  - 'Enable'
  - Filename 'Space parse bitstream'
  - Then click 'Parse Bitstream'
- Click the 'Frame Range' tab: select frames 1 to 3 (i.e. just the first 3 frames).
- Click 'OK' at the bottom of the 'Frame range' tab.

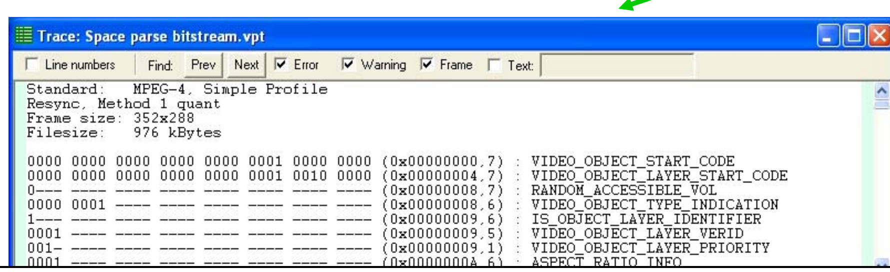




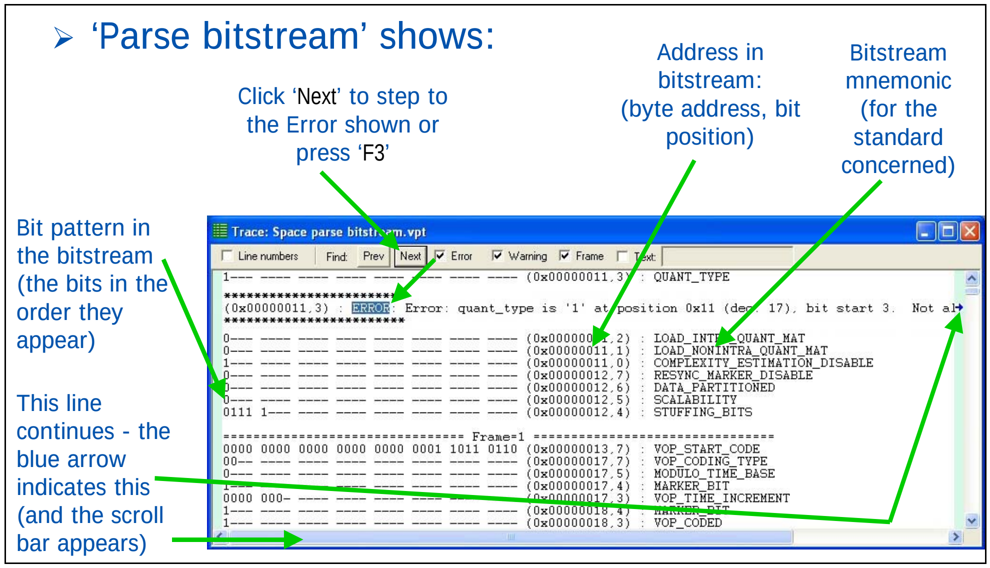
Trace - parse bitstream

- Single step through the first 4 frames of the video (using 'Ctrl+A' or the Pause/Step forward icon) (Frame number is at the top: click continue when the error pops-up)
- Click 'View trace' (or Ctrl+V)

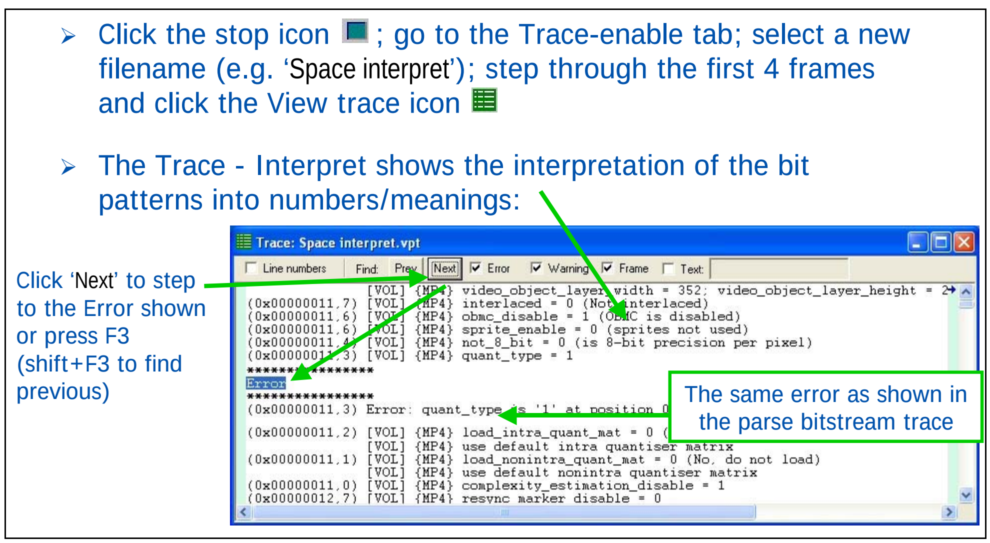




Trace - parse bitstream (2)

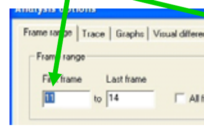


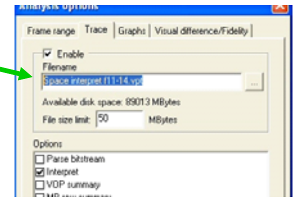
Trace - interpret



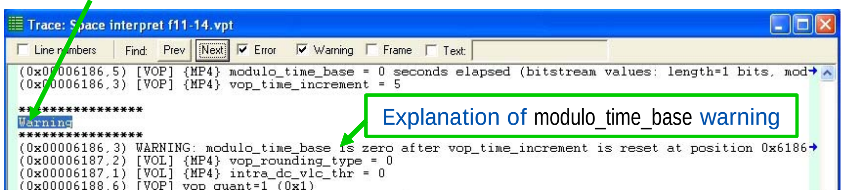
Trace - interpret (2)

- Repeat but look at frames 11-14: (new filename).
- Deselect ‘Frame’; click ‘Next’ (or F3).





Warning highlighted



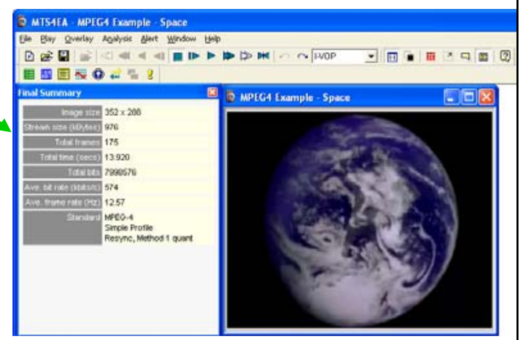
Note: the ‘modulo_time_base’ field is actually at 0x6186,5 but MTS4EA cannot know it is incorrect until the value of ‘vop_time_increment’ occurs at 0x6168,3 so this is the address given. *This can happen with many errors: it is only a later field which shows that the earlier field was incorrect*

Trace files result

- The Trace files have shown that the bitstream has:
  - an incorrect value for quant_type - Method 1 quantisation has been selected, but this is not allowed in MPEG-4 Simple Profile
  - errors in the modulo_time_base field
- Note that the Trace files can be very large if Trace is active over many frames - that is why it is suggested always to use a range of frames

'MPEG-4 Space' Summary Tooltip

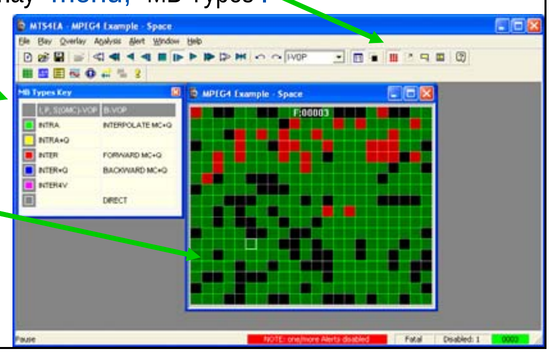
- Switch on Summary tooltip: icon or Ctrl+U or 'Overlay' menu 'Summary tooltip' (to force undock, hold <Ctrl> and drag with mouse)
- The Final Summary Tooltip at the end gives:



  - Picture size: 352 pixels High x 288 pixels Wide (CIF)
  - File size is 976K Bytes
  - There are 175 frames
  - Play time of the sequence is total 13.920 seconds
  - Total bits: 7,998,576
  - The bandwidth required to transmit this is 574 kbits/second
  - The frame play rate is 12.57 frames per second (Hz)
  - The clip is MPEG-4 Simple Profile with Resync markers & Method 1 quant

There are syntax errors - but is it optimal?

- The sequence looks ~OK - now see if the codec makes best use of the MPEG-4 standard?
  - Click the 'X' at the top right of the Final Summary window to close it.
  - Click or Ctrl+Y or 'Overlay' menu, 'MB Types'.
  - The MB types color key appears.
  - Step forward 3 frames: the MB types are now color-coded.

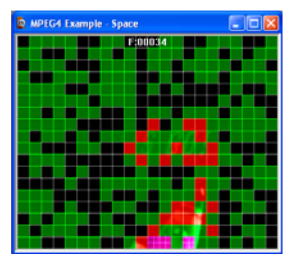


MB types overlay color key

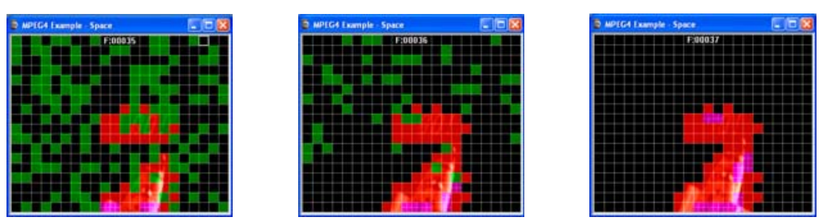
- The MB types key can be
  - Docked to one edge.
  - Undocked.
    (*to force undocking, hold the <Ctrl> key while dragging with the mouse*)
  - Resized.
  - Switched off altogether by clicking the ‘X’ (top right) (to display it again, click the MB types icon twice).

‘MPEG-4 Space’ - is it optimal?

- Step forward quickly to frame 34
  - There are lots of **green** (Intra) coded MacroBlocks in the static black background (use most bits, generally).



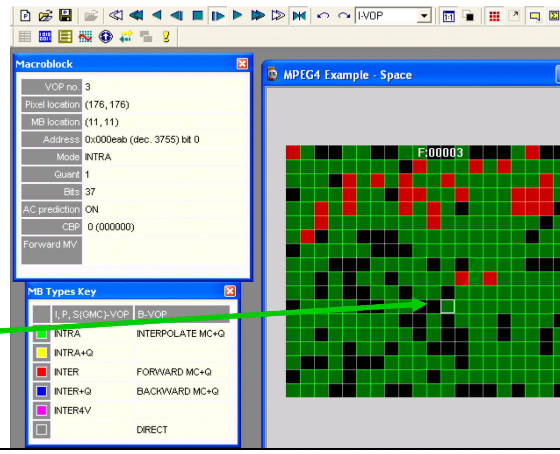
- Now step forward slowly over frames 35-37:



- ...The Intra MacroBlocks disappear.

MB tooltip (MB=MacroBlock)

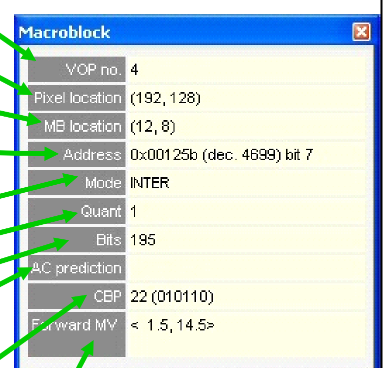
- Now click stop (the icon); switch on the MB tooltip by clicking the icon and step forward to frame 3
- The MB tooltip is displayed:
  - This provides detailed information on the MacroBlock indicated by the white box highlight



MB tooltip: information provided

- Information provided by the MB tooltip:

  Note: this varies substantially by video standard - see 'How to use MTS4EA' section of the User Manual.
  - Current display VOP/frame
  - Location of the top-left pixel of the highlighted MacroBlock
  - MB location in X,Y coordinates, where 0,0 is top-left
  - Address of the start of the MB data in the bitstream
  - Encoding mode (Intra, Inter, etc.)
  - Quantizer used in the MB
  - Number of bits used in the MB
  - Whether AC prediction is on or not
  - Coded Block Pattern (Y0, Y1, Y2, Y3, U and V)
  - Motion Vectors (X, Y)



'MPEG-4 Space' - is it optimal? (2)

- Stop the sequence; switch on the Summary tooltip and look at the Intra MBs in frames 1-3:
  - the Intra MBs in frame 1 typically take 22 bits (an I-VOP)
  - the Intra MBs in frames 2 and 3 (both P-VOPs) typically take 27 or 37 bits
- Forward on to frame 37:
  - the black background areas are 'not_coded' (i.e. they take no bits, except for the 'not_coded' flag bit)
- Clearly there is a major difference:
  - when the whole frame is almost completely black, it seems that the encoder chooses many Intra MBs
- ***Why??***

'MPEG-4 Space' - making it optimal

- It looks like the encoder is choosing MB types non-optimally - it should be 'not_coded' MBs for the background for all the P-VOPs: this is wasting many bits.
  - Are there software bugs, where the encoder is too sensitive to minor changes in the grey level?
  - To do: check the encoder part of the codec that decides when to use Intra MBs, Inter MBs and not code the MB.
  - Fixing this problem would save many wasted data bits.

  Note for information: the MPEG-4 standard provides more data bits for Intensity - grey level - than for color information, so it is naturally more sensitive to changes in grey level (this mimics the human visual system). However, it seems in this example that the sensitivity to grey level is too great.

- ***MTS4EA hasn't fixed the problem, but it has told you exactly where to look.***

Tutorial 4: MPEG-4 ASP Window Car

This tutorial focuses on:

- Video navigator view
- Use of Trace
  - Parse bitstream
  - Interpret
- Common error: stuffing_bits
- Synchronizing views
- Saving setup in project files

It uses the supplied video clip in the 'File' menu: ‘Example files’, 'MPEG-4 Elementary Streams', ‘Window Car’.

(There are buffer conformance errors also, but these are ignored in this tutorial.)

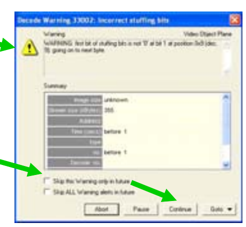
NOTE. *In this tutorial it is assumed that you have gone through the previous tutorials (as indicated on the next slide).*

Prior tutorials assumed ...(1, 2 and 3)

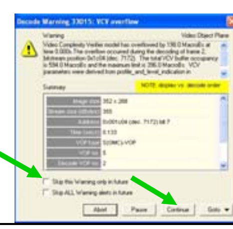
- It is assumed that you have done the previous tutorials and know how to:
  - Play/Stop/Advance/Fast fwd/blind fast fwd the video.
  - Use the alert pop-ups, interpret them and understand how to click 'Skip' and 'Continue'.
  - Use the MacroBlock and Summary tooltips and understand the information provided.
  - Use the MacroBlock overlays: MB types, motion vectors and intra coding frequency.
  - Set the frame range.
  - Set the Trace file name and options.
  - View the Trace files: Parse bitstream and Interpret.

Common error: stuffing_bits

- Go to 'Example file' 'MPEG-4 Elementary streams' 'Window Car' - 'stuffing_bits' error occurs when the file is loaded, at stream address 0x9, bit 1.

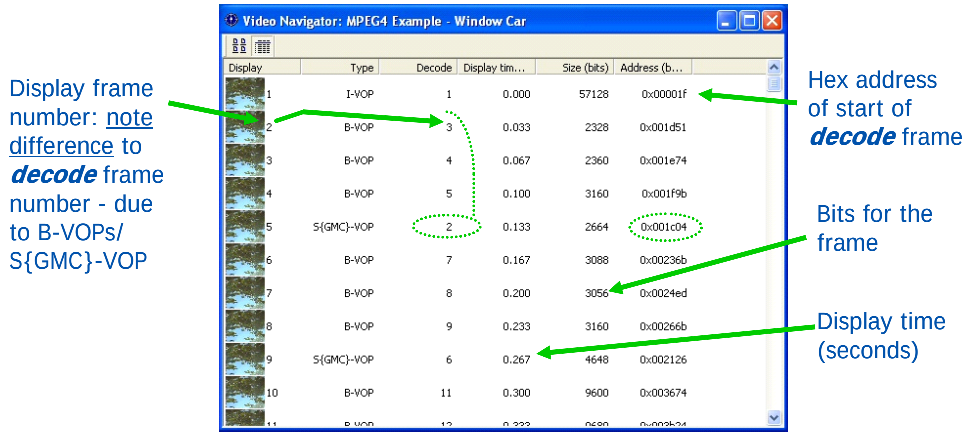


- Click 'Skip this..', 'Continue' when the 'stuffing_bits' alert pops up.
- Play the video to the end (click 'Skip this..', 'Continue' when the VCV overflow alert occurs).



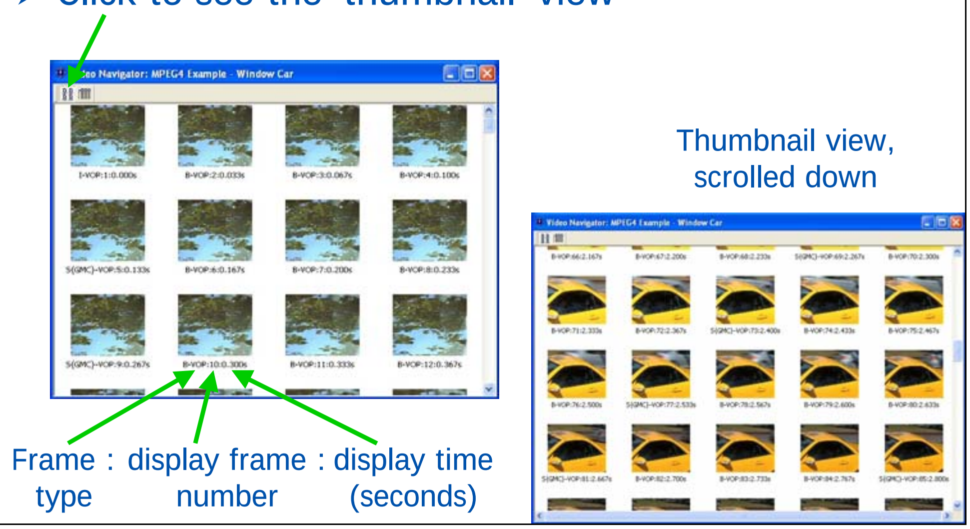
Video navigator view

- Click the video navigator icon (or 'Play' menu, 'Video navigator').
- The video navigator window appears, either in 'Detail' view (below) or 'Thumbnail' view (next page):



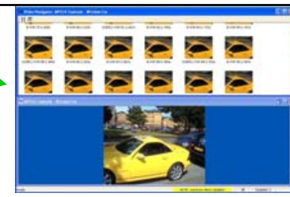
Video navigator view (2)

- Click to see the 'thumbnail' view

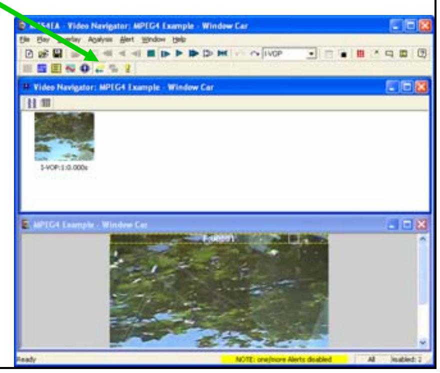


Synchronize video frame and navigator

- On the 'Window' menu, click 'Tile horizontally'.

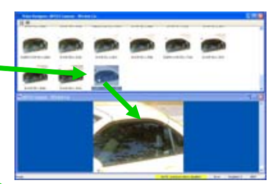


- Click the 'Synchronize views' icon
  - In this case, the displays change to the first frame: this is because the end of the sequence had been reached and therefore frame 1 was the first that could be synchronized.
  - Play through the whole sequence again: the video navigator and the video views are synchronized.



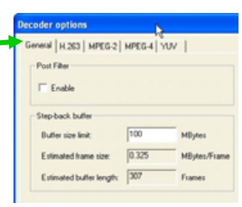
Jump to video frame - cached data

- The last frame is now displayed in the video window
  - Scroll back up the navigator view and double-click on frame 57
  - The decoding restarts from the beginning and stops at frame 57.



  - Double-click on frame 51: the view changes immediately - it does not start from the beginning again.
  - The reason is that the data is held in a cache.

    - MTS4EA has a cache of data: the size is selected in the 'Play' menu, 'Decoder options', 'General' tab.
    - BUT if the video has been stopped (e.g. played to the end) then the cache must be refilled.



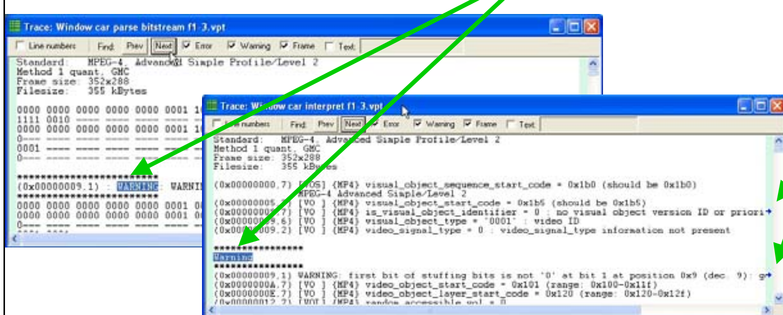
Synchronized views

- Many other views can be synchronized with each other:
  - Video window
  - Video navigator
  - Buffer analysis
  - HexView
  - Alert log
  - Fidelity analysis
  - Trace / Parse Bitstream and Trace / Interpret

- To go to the same place in another view.
  - Ensure 'Synchronize views' is on (icon pushed in).
  - In any view, right-click, select 'Goto view' and select the view from the drop-down menu.

Look again at 'stuffing_bits' alert: Trace

- View the Trace files frames 1-3: the 'Warning' alert is shown

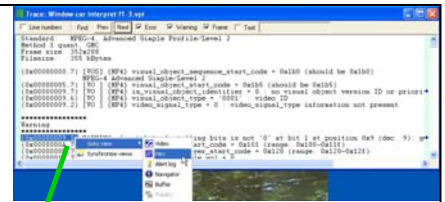


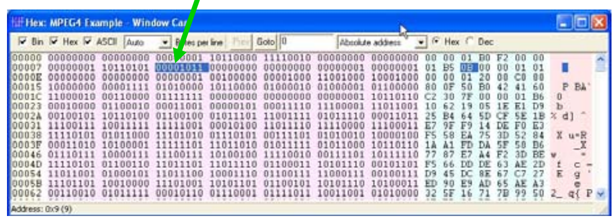
The blue arrows indicate the line is longer than the window width - use the scroll bar to see the rest of the line

Note: the warnings, errors and fatals are *always* shown in the Trace / Parse bitstream and Interpret files, irrespective of whether or not the pop-up alerts are enabled

Look at the bitstream data - 'HexView'

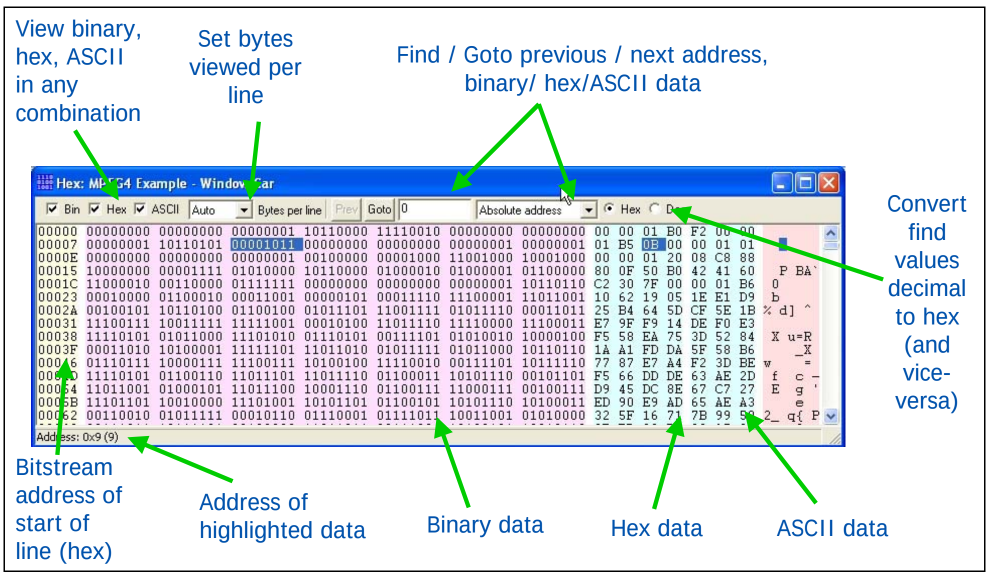
- In the Trace / Interpret window, select the address (0x00000009,1) then right-click; select 'Goto view', 'Hex':
  - The 'HexView' is opened with the same location highlighted.
  - It can be seen that bits 1 and 0 of byte 9 are both 1: but for stuffing bits, the bits should be '01'.





MTS4EA has shown exactly where the problem is.

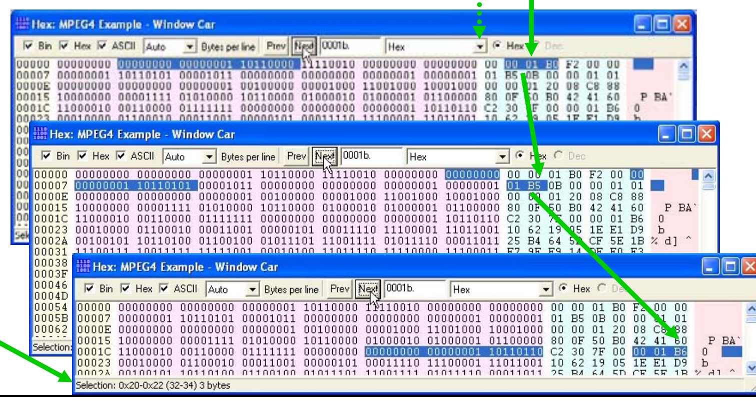
The HexView window



HexView ‘Find’ using ‘.’ wildcard

- You can also search using a ‘.’ as a wildcard character, e.g. Find ‘0001b.’ - select ‘Hex’ to find ‘0001b0’, ‘0001b5’ and ‘0001b6’ (i.e. MPEG-4 start codes):

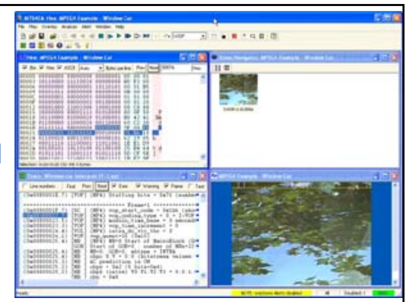
(See ‘How to use MTS4EA’ section in the User Manual for more information on HexView)

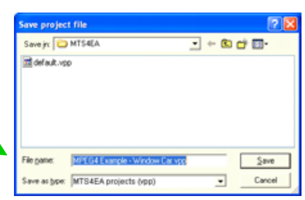


Addresses and number of bytes found

Saving status with Project files

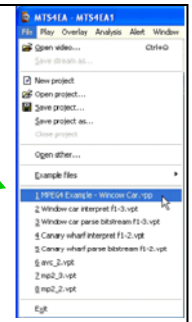
- The Project file allows the status of *most* of the windows and setups to be stored:
  - For example, suppose this is the desired setup of the MTS4EA windows: *this setup can be stored to save time when restarting working with MTS4EA.*
  - On the ‘File’ menu, select ‘Save Project...’ or click the icon.
  - A filename is provided (the name of the Elementary Stream video file with the ‘.vpp’ extension appended) - A different name can be entered.



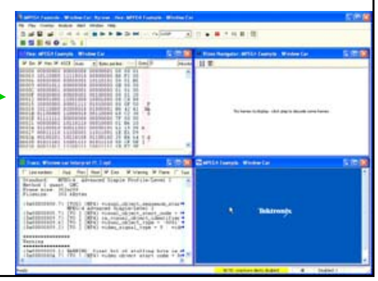


Restoring status with Project files

- Close MTS4EA completely, then restart it and load the project file from the file menu: the project file is listed under the recent files.



- MTS4EA reopens with the same video file, window setup etc.
  - Note: the alert pops up when the video file is loaded: click 'Skip this' and 'Continue', and the restored display is:



Conclusions

- It is easy to see many different views on the video data, and navigate between them.
- *Using these different views, MTS4EA has shown exactly where the 'stuffing bits' problem is.*
- Project files can be used to store setups and save time when stopping and restarting work.

Tutorial 5: MP4 File Packet Woman

This tutorial focuses on:

- Opening an MP4 file and viewing the tracks contained
- Syntax error: not obeying Profile/Level restrictions
- Extracting/saving the video track
- Viewing MP4 file structure

It uses the supplied video clip in the 'File' menu: 'Example files', 'MP4 files', 'Packet Woman'.

(There are buffer conformance errors also, but these are ignored in this tutorial.)

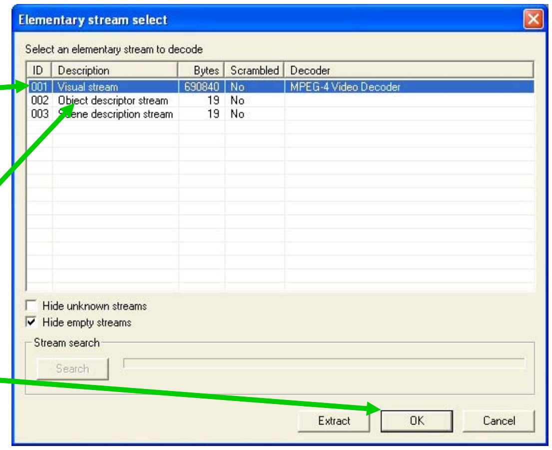
Note: *In this tutorial it is assumed that you have gone through the previous tutorials (as indicated on the next slide).*

Prior tutorials assumed …(1 - 4)

- It is assumed that you have done the previous tutorials and know how to:
  - Play / stop / advance / fast fwd / blind fast fwd the video.
  - Use the alert pop-ups and interpret them.
  - Use the MacroBlock and Summary tooltips and understand the information provided.
  - Use the MacroBlock overlays: MB types, motion vectors and intra coding frequency.
  - Set the frame range.
  - Set the Trace file name and options.
  - View the Trace files: Parse bitstream and Interpret.
  - Use the hexview bitstream viewer.

Tracks in an MP4 file

➢ Click 'File' menu, 'Example files', 'MP4 files', 'Packet Woman': the dialog box listing the tracks in the MP4 file appears:



Video track, ID=001 there can be multiple video tracks: the first video track is selected automatically.

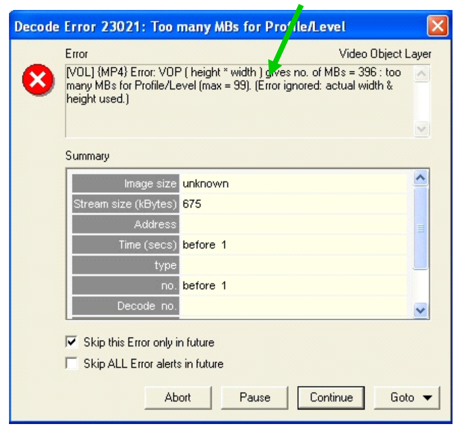
Other tracks with 'meta' data (unsupported in MTS4EA)

Click 'OK' (Note: 'Extract' is later in this tutorial).

Video track - 'Level' conformance error

➢ The video track is now analyzed - an error alert:

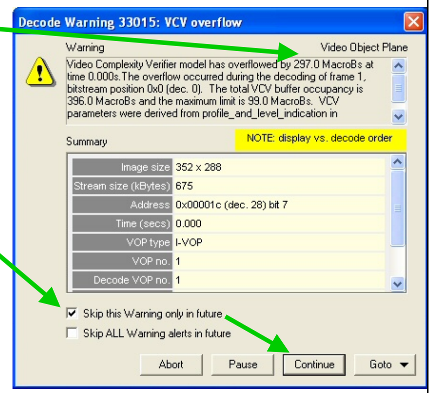
- The VOP has more MacroBlocks (396) than allowed in MPEG-4 Simple Profile / Level 1: the maximum number allowed is 99.
- This is probably an error with the 'Level' - it is stated as 'Level 1' in the encoded file, but it is actually the size of a 'Level 2'.



- Click 'Skip this...' and MTS4EA can continue anyway (using the actual size).

Buffer conformance warning

- There is also immediately a VCV overflow warning
  - overflow by 297 MBs
  - this is to be expected, as the Level is incorrectly given as L1 (which has a limit of 99 MBs) whereas it <u>should</u> be L2 (which has a limit of 396 MBs)

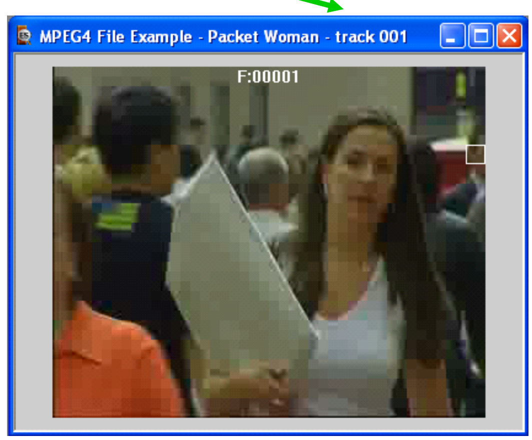


- For now, just click 'Skip' and 'Continue' to ignore for now

<u>Buffer conformance analysis is covered in detail in Tutorial 9 - see this tutorial for more information</u>

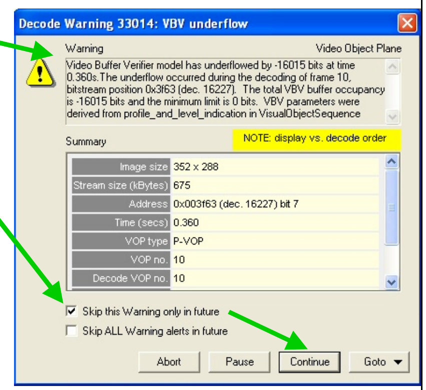
Video track played ...

- The track ID is at the top of the video window



Another buffer conformance warning

- There is also a VBV underflow warning during decode of frame 10
- For now, just click 'Skip' and 'Continue' to ignore this error also



Buffer conformance analysis is covered in detail in Tutorial 9 - see this tutorial for more information

View MP4 file structure

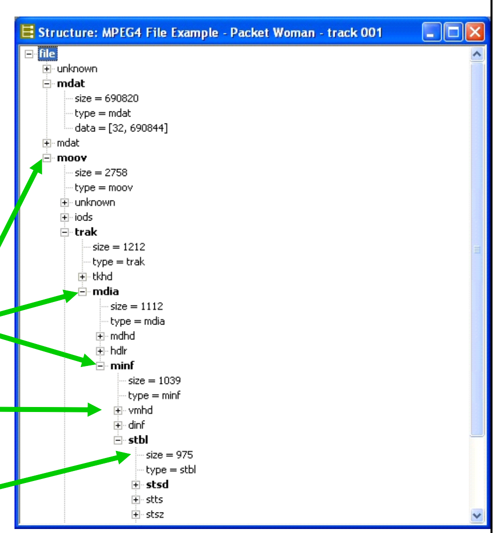
- Make the video window active (click its title bar)
- Click 'Analysis' menu, 'View file structure' (or Ctrl+R) or click the icon

Top-level atoms in the MP4 file

Video-related atoms in **bold**

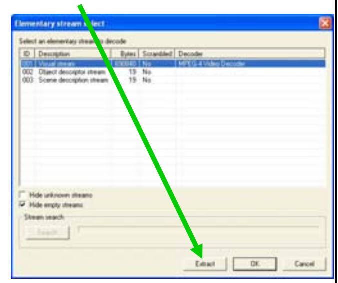
Click to '**+**' next to each atom name to expand it (changes to '**-**' when fully expanded)

Data for the atom

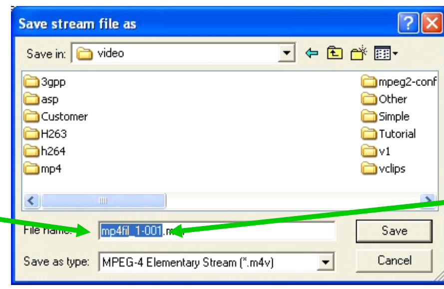


To save the '.m4v' video from MP4 file

- Close the video window, then reopen it: when the tracks list appears, this time click 'Extract':



- The 'Save stream as' dialog box opens:



Enter different file name if required

Track ID and correct video file type has been automatically appended to file name (you can amend this)

Note: 'File' menu 'Save stream as…' can be used to open the above dialog box when a video track has been opened from an MP4 file but was not extracted at the time the MP4 file was first opened

MTS4EA has …

- Shown which tracks are in an MP4 file.
- Extracted and played the video.
- Shown that there is a 'Level' conformance error, and precisely what it is and where it is.
- Shown that there are two buffer conformance errors *(see Tutorial 9 for details of buffer analysis).*
- Nevertheless allowed the video to be played and checked for other errors.

Tutorial 6: MP4 File Piccadilly Circus

This tutorial focuses on:

- Opening an MP4 file; viewing the video track
- Using the Summary tooltip and MB types overlay
- Quick, straightforward checks for optimization that could easily *reduce bit usage by 45%* in this sequence

It uses the supplied video clip in the 'File' menu: ‘Example files’, 'MP4 files', ‘Piccadilly Circus’.

(There are two buffer conformance errors also, but these are ignored in this tutorial.)

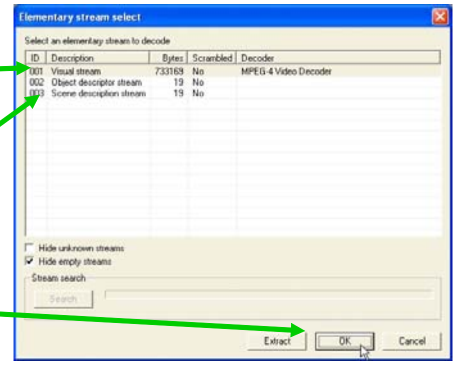
Note. *In this tutorial it is assumed that you have gone through the previous tutorials (as indicated on the next slide).*

Prior tutorials assumed ...(1 - 5)

- It is assumed that you have done the previous tutorials and know how to:
  - Play / stop / advance / fast fwd / blind fast fwd the video.
  - Use the alert pop-ups and interpret them.
  - Use the MacroBlock and Summary tooltips and understand the information provided.
  - Use the MacroBlock overlays: MB types, motion vectors and intra coding frequency.
  - Set the frame range.
  - Set the Trace file name and options.
  - View the Trace files: Parse bitstream and Interpret.
  - Use the HexView bitstream viewer.
  - Open MP4 files, view/extract the video, view the MP4 file structure.

Extract the video from the MP4 file

- Click 'File' menu, 'Example files', 'MP4 files', 'Piccadilly Circus': the dialog box appears listing the tracks in the MP4 file:

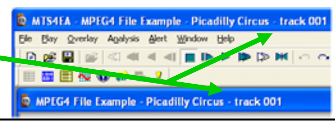


Video track, ID=001 there can be multiple video tracks: the first video track is selected automatically

Other tracks with audio and 'meta' data (unsupported in MTS4EA)

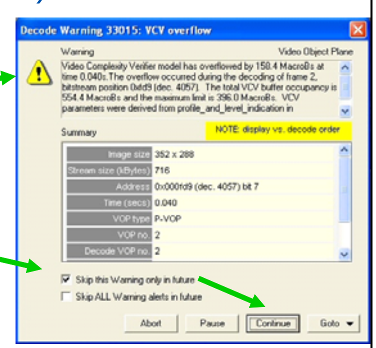
Click 'OK'

Track number shown



Begin …

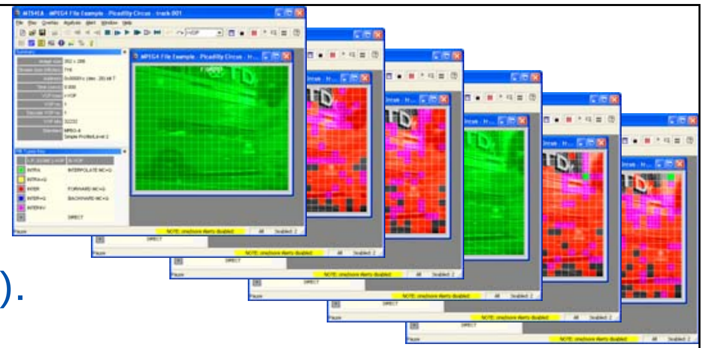
- Before playing:
  - Switch on the Summary tooltip (or Ctrl+U)
  - Switch on the MB types overlay (or Ctrl+M)
  - Then click 'Play' or 'Pause/Step forward' through the sequence to the end (/ Ctrl+F, or / Ctrl+A).
- A VCV buffer conformance error occurs while decoding frame 2.
- Ignore this for now - click 'Skip' and then 'Continue'.



Buffer analysis is covered in Tutorial 9

Continue ...

- The first frame is green, as is every 3rd frame (i.e. all Intra MBs).



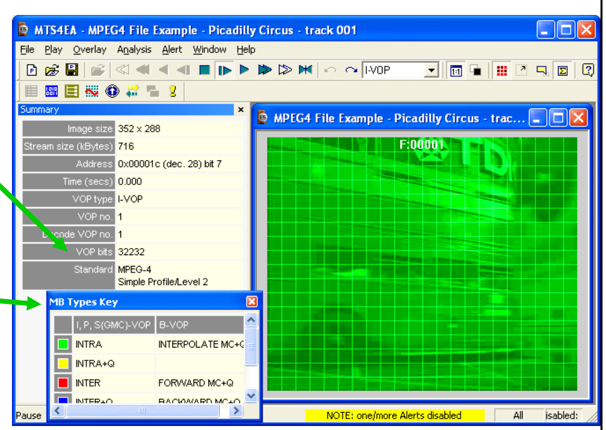
- Another type of buffer conformance error occurs - this time VBV underflow - at frame 34, but also click to 'Skip this' and 'Continue'. (see Tutorial 9 for buffer analysis)

Look more closely at the VOP types

- Replay the video file but using Pause/frame advance, step to the first frame.

- The first frame takes 32,456 bits.

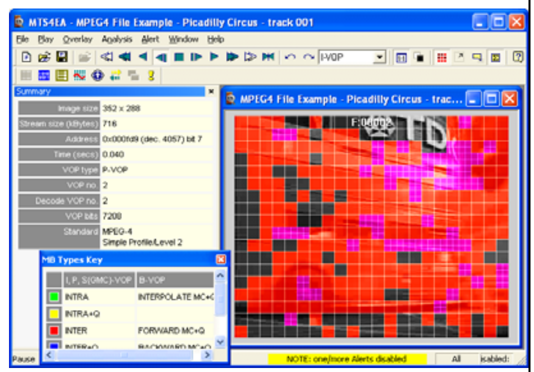
(Click the 'X' to remove the MB types color key, if you wish - click twice to redisplay it; hold the<Ctrl> key to force undocking while dragging it with the mouse.)



Step through the video ...

- Using Pause/frame advance, step through the next 6 frames:

Frame number	Frame type	Bits used
2	P-VOP	7,208
3	P-VOP	7,792
4	I-VOP	33,152
5	P-VOP	7,216
6	P-VOP	8,776
7	I-VOP	34,344



(frame 2 shown above)

Optimize the choice of VOP types

- The I-VOPs take ~4.5 times as many bits as the P-VOPs.
- ***Why are there I-VOPs every 3rd frame?***
- There is probably no need to have I-VOPs so often - if the frequency of I-VOPs were reduced to 1 in 30, ***the bits used for the sequence would reduce by ~45%.***
- ***MTS4EA has quickly shown how to easily reduce substantially the bits used.***

Tutorial 7: 3GPP File Mobile Hands

This tutorial focuses on:

- Opening a 3GPP file and viewing the tracks contained
- Syntax error: reserved Profile/Level indicator
- Quick, straightforward checks for optimization - may or may not be applicable in a wireless environment

It uses the supplied video clip in the 'File' menu: ‘Example files’, '3GPP file', ‘Mobile Hands’.

Note: *In this tutorial it is assumed that you have gone through the previous tutorials (as indicated on the next slide).*

Prior tutorials assumed ...(1 - 6)

- It is assumed that you have done the previous tutorials and know how to:
  - Play / stop / advance / fast fwd / blind fast fwd the video.
  - Use the alert pop-ups and interpret them.
  - Use the MacroBlock and Summary tooltips and understand the information provided.
  - Use the MacroBlock overlays: MB types, motion vectors and intra coding frequency.
  - Set the frame range.
  - Set the Trace file name and options.
  - View the Trace files: Parse bitstream and Interpret.
  - Use the HexView bitstream viewer.
  - Open MP4 files, view/extract the video, view the MP4 file structure.

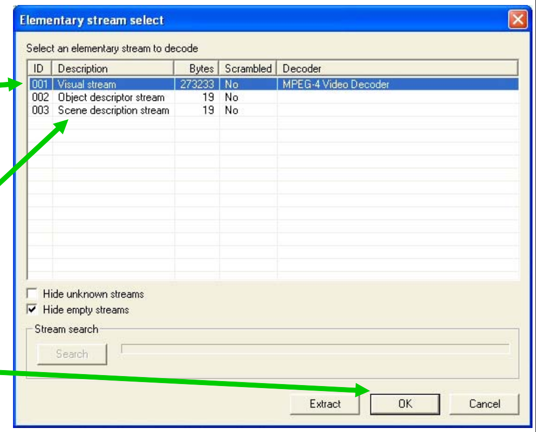
Tracks in a 3GPP file

- Click ‘File’ menu, ‘Example files’, ‘3GPP file’, ‘Mobile Hands’: the dialog box listing the tracks in the 3GPP file appears:

Video track, ID=001 there can be multiple video tracks: the first video track is selected automatically

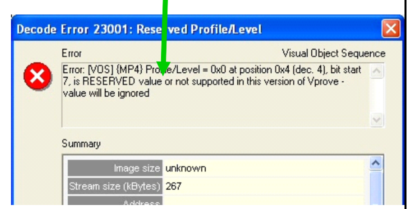
Other tracks with ‘meta’ data (unsupported in *MTS4EA*)

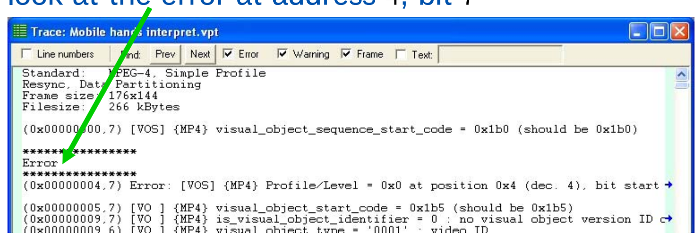
Click ‘OK’
(see Tutorial 5 regarding Extract’)



Alert on syntax error …

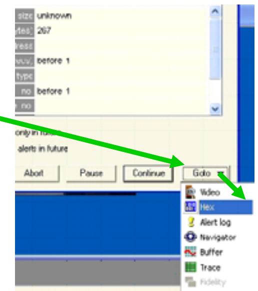
- Reserved value (0) for ‘profile_and_level_indication’:
  - A very common error in mobile video streams.
- Click ‘Continue’ (do *not* enable ‘Skip’)
- Run Trace/Interpret (frame 1):
  - look at the error at address 4, bit 7

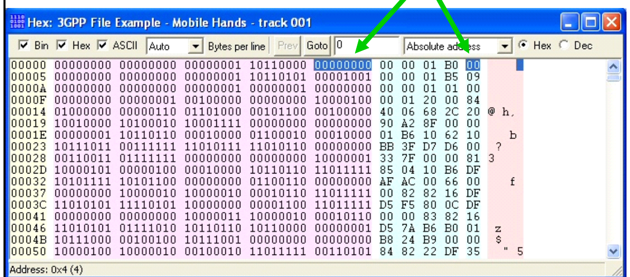




Look at error bits in bitstream …

- Play video again, and at the error click 'Goto', 'Hex'.
- The HexView appears, with location of error bits highlighted:



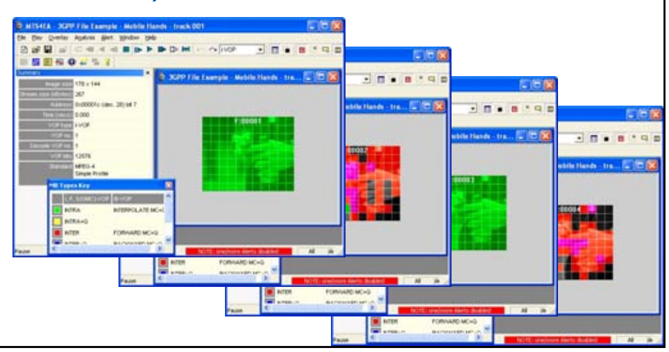


The HexView can also be viewed by selecting the address in Trace / Interpret *or* Trace / Parse Bitstream*, then right-click and select* 'Goto', 'Hex'.

- MTS4EA has shown precisely where the problem is.

Play the video - look at the VOP types

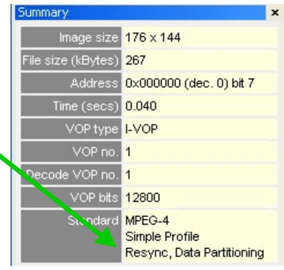
- Stop the stream (if playing)
  - Switch on the Summary tooltip (or Ctrl+U)
  - Switch on the MB types overlay (or Ctrl+M)
  - 'Play' or 'Pause/Step forward' through the sequence to the end (/ Ctrl+F, or / Ctrl+A)
- The first frame is green, as is every second frame; (that is, all Intra MBs).



Error resilience? Optimize by ~15%?

- Switch on the Summary tooltip (or Ctrl+U);
  Pause/Step forward to frame 1 (or Ctrl+A)

- In a mobile environment, error resilience is vital
  - 'resync markers' and 'data partitioning' are both **ON** (good for error resilience)



- So, with these error resilience tools being used, could the frequency of I-VOPs be reduced to say 1 in 5??
  (instead of the current 1 in 2)

- Doing this would ***reduce the bit-rate by ~15%***
  (or allow significantly better quality video in the same bit rate)

Tutorial 8: H.264/AVC Byte Stream Canary Wharf

This tutorial focuses on debugging an H.264/AVC stream:

- Opening an H.264/AVC byte stream
- Syntax error: trying to use an invalid (un-initialized) Picture Parameter Set
- Finding the frame number of the error
- Examining the H.264/AVC byte stream syntax using Trace/Parse bitstream and Trace/Interpret and finding the error

It uses the supplied video clip in the 'File' menu: ‘Example files’, 'H.264/AVC Byte Stream', ‘Canary Wharf’.

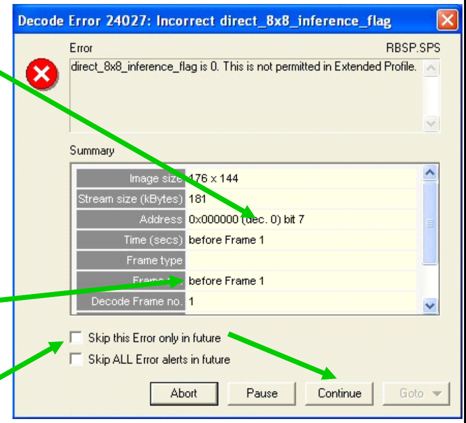
Note: *In this tutorial it is assumed that you have gone through the previous tutorials (as indicated on the next slide).*

Prior tutorials assumed …(1 - 5)

- It is assumed that you have done the previous tutorials and know how to:
  - Play / stop / advance / fast fwd / blind fast fwd the video.
  - Use the alert pop-ups and interpret them.
  - Use the MacroBlock and Summary tooltips and understand the information provided.
  - Use the MacroBlock overlays: MB types, motion vectors and intra coding frequency.
  - Set the frame range.
  - Set the Trace file name and options.
  - View the Trace files: Parse bitstream and Interpret.
  - Use the HexView bitstream viewer.

Syntax error in Canary Wharf byte stream

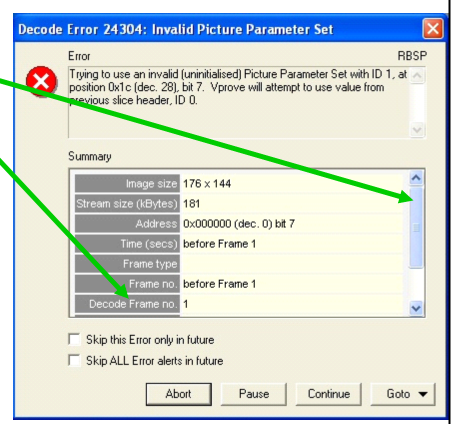
- Go to 'Example file' 'H.264/AVC Byte Streams' 'Canary Wharf' - incorrect 'direct_8x8_inference_flag' error occurs when the file is loaded, at stream address 0x0, bit 7



- In which frame is the syntax error?
  - The time says "before Frame 1" : this is the *displayed* frame number.
- Click 'Skip this', 'Continue'.

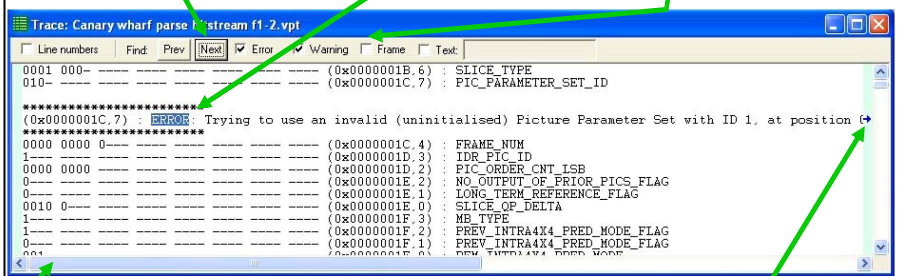
Another syntax error …

- An invalid Picture Parameter Set ID error also occurs during the decode of Frame 1.
- Use the scroll bar to see the rest of the information on the error.
- Frame number 1 being decoded.
- Run 'Trace' / 'Interpret' and 'Trace' / 'Parse bitstream' on frames 1 to 2 ('Analysis' menu, 'Trace enable' and 'Frame range' set range 1 to 2)



‘Trace’ - ‘Parse bitstream’

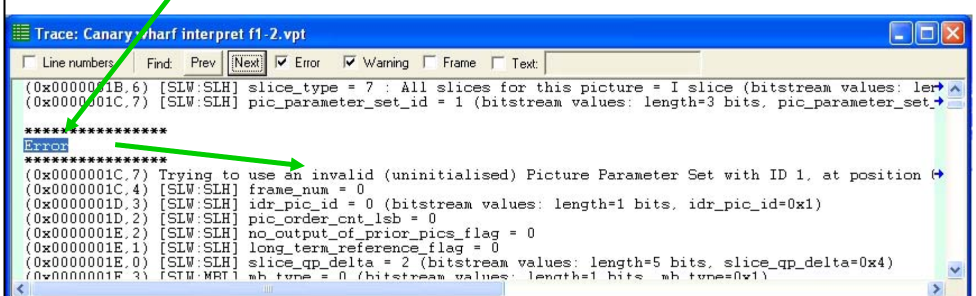
- View the Trace file: clear the ‘Frame’ box and click ‘Find Next’ twice: the Error alert is shown:



The blue arrow indicates the line is longer than the window width - use the scroll bar to see the rest of the line.

‘Trace’ - ‘Interpret’ - **look at the error**

- View the Interpret trace (same frame range: 1-2); clear the ‘Frame’ box and click ‘Find: Next’ twice
- The ‘Error’ alert is shown:

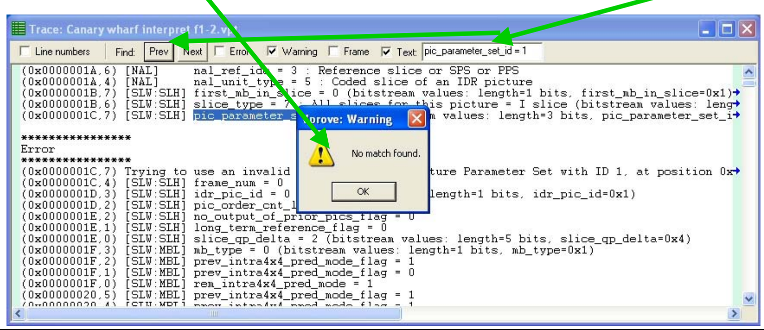


‘Trace’ - ‘Interpret’ - what is the error?

- A slice header identifies which Picture Parameter Set is used for the current slice.
- This refers to ‘pic_parameter_set_id’ number 1 - *but has this been set?*
- To find out, search for ‘pic_parameter_set_id =’ in the Interpret trace file already open (frames 1-2).
  - select ‘Text’ box
  - enter ‘pic_parameter_set_id =’

Find 'pic_parameter_set_id' is 1 in frames 1-2

- Search for ‘pic_parameter_set_id =’ earlier: use ‘Find Prev’ (i.e. previous occurrence, before the error)
- It does not occur before the location of the PPS ID error (0x1c,7)



Conclusion: Picture Parameter Set ID error

- ‘pic_parameter_set_id = 1’ does not occur before the slice header tries to use this in frame 1 - clearly an error.
- With this error, the MTS4EA decoder uses the data from the last known good ‘pic_parameter_set_id’ (ID 0) and continues decoding.
  (note: this may not always be possible and could generate other consequential errors)
- The error is in setting the PPS ID to 1, at stream address 0x1c, bit 7.
- ***MTS4EA has shown exactly <u>what</u> the error is and exactly <u>where</u> the error is.***

Tutorial 9: Buffer Analysis in MPEG-4 and H.264/AVC

This tutorial focuses on:

- Buffer analysis in real-time (VBV, VCV, VMV of MPEG-4 ES)
  - Information provided - values, overflow/underflow indications
  - Icon controls for navigation through the graphs
- Buffer conformance errors/alerts
  - In the Graph window
  - Alert pop-ups
  - In Trace files
- Achieving conformance
  - Altering the buffer parameters and seeing the effect in the graph
  - Adjusting values to prevent overflow
- HRD buffer analysis in H.264/AVC

It uses the supplied video clips in the 'File' menu: ‘Example files’, 'MPEG-4 Elementary streams', ‘Man Walking’ and the H.264/AVC example file 'Bus Junction'

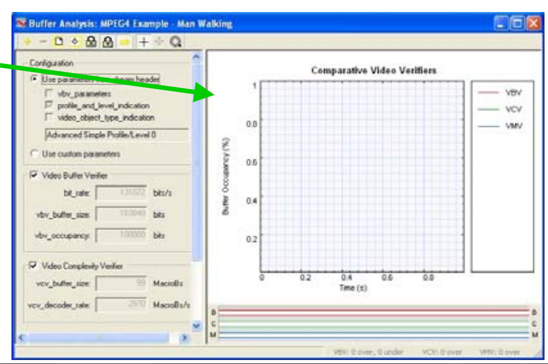
Note*: In this tutorial it is assumed that you have gone through the previous tutorials (as indicated on the next slide).*

Prior tutorials assumed …(1 - 3)

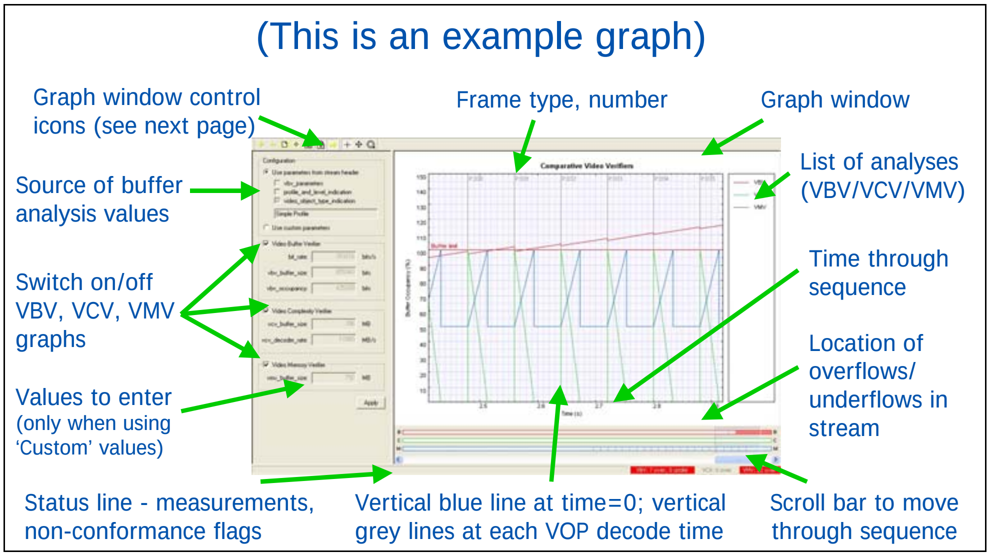
- It is assumed that you have done the previous tutorials and know how to:
  - Open example streams.
  - Play / stop / advance / fast fwd / blind fast fwd the video.
  - Set the frame range.
  - Set the Trace file name and options.
  - View the Trace files: Parse bitstream and Interpret.
  - Use the alert pop-ups and interpret them.
- It is also assumed that you *already* understand the principles of buffer analysis/conformance (VBV, VCV, VMV) in MPEG-4 and H.264/AVC (HRD): MTS4EA does *not* teach this.

Begin buffer analysis…

- Open the Example MPEG-4 Elementary Stream 'Man Walking' (do not play it yet)
- Click 'Analysis' menu, 'View buffer analysis...' or click icon : the Buffer Analysis window appears



The buffer analysis window



Buffer analysis - icon controls

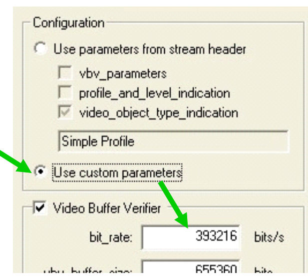
Icon	Function
+ −	Zoom in (+) and zoom out (-) centered on the center of the window - affected by the 'Lock X / Y' icons (see below)
	Fit all data into window
	Goto origin (zero) i.e. start of video sequence
	Lock X/Y zoom in/out and scrolling/panning. E.g. when the 'Lock Y' button is pressed, zoom in 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; i.e. fill the analysis data in real-time, as the video is being decoded and scroll the window to the right
+	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) - this is affected by the 'Lock X-Y' icon - see above
	Zoom in/zoom out centered on the location of this cursor - this is affected by the 'Lock X-Y' icon - see above. Press the <shift> key to zoom out

Buffer conformance values

- MTS4EA takes VBV, VCV and VMV values from
  - The parameters specified in the stream - or if not there, from:
  - The implied values from the Profile/Level, as given in the standard, or if there is no 'profile_and_level_indication':
  - The implied values from the Object Type (Simple or Advanced Simple) at the highest Level in each.

  or

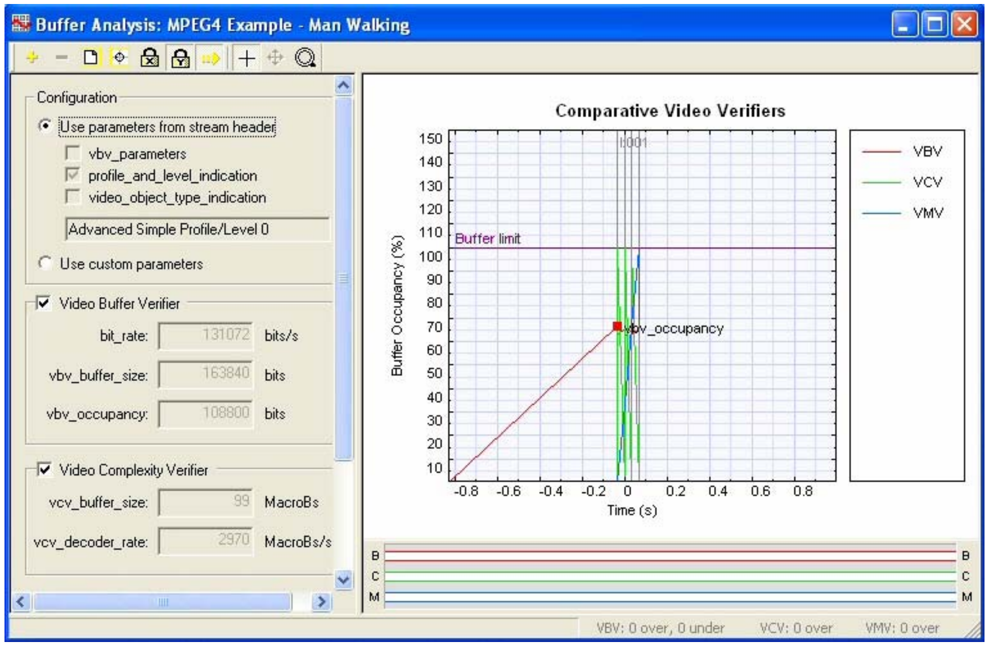
  - You can enter them yourself:



Step through the video

- Click 'Pause/Step forward' 3 times (or or Ctrl+A) - *nothing appears in the window*
- The reason is that the 'Man Walking' sequence contains B-VOPs and the buffer analysis cannot start until there is sufficient data for the calculations - in this case, not until frame 4
- ...so, click 'Pause/Step forward' 1 more time (or or Ctrl+A) and the graphs appear
- Click 'Fit to window' icon to see view on next page

First 4 frames

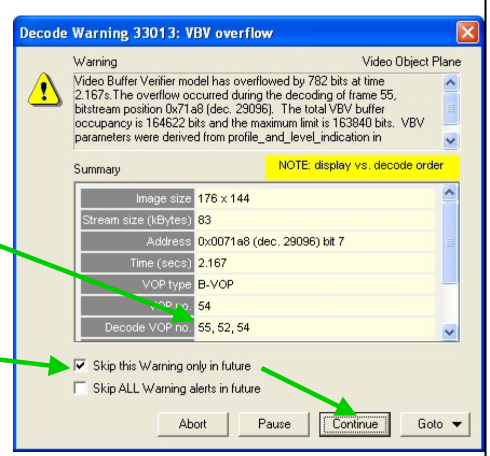


Continue …

- Click 'Blind Fast Fwd' (or ▷ or Ctrl+B) - the graphs build up but a pop-up error alert occurs

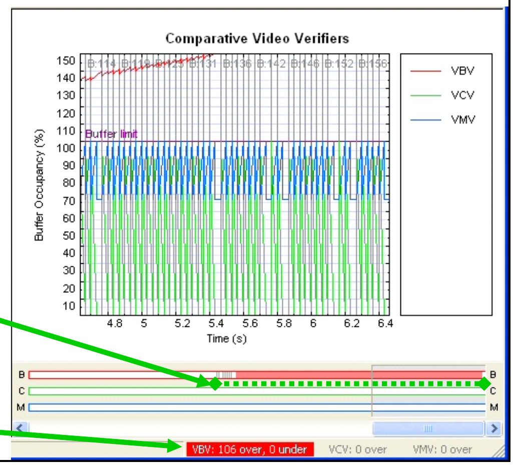
- A VBV overflow has occurred, *decoding* frame 55 (display frame number is 54)

- 'Skip' to ignore and 'Continue'



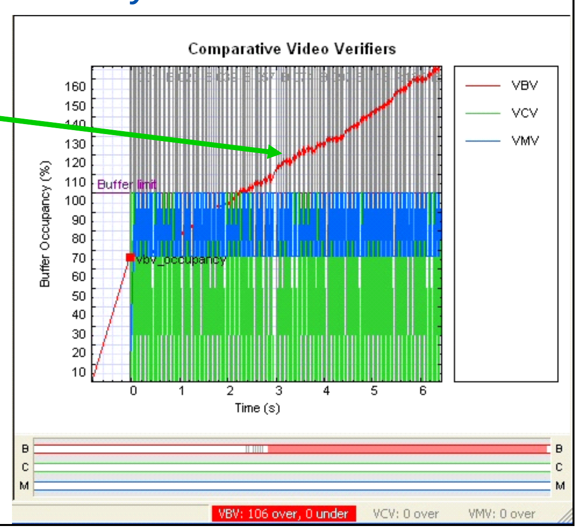
Buffer analysis of whole stream

- The red VBV line goes up, above the 'Buffer limit' line and goes off the top of the graph view.
- When the decoding has stopped the graphs look like:
- Part of stream with VBV overflow:
- No. of frames with VBV overflow:



View the whole buffer analysis

- Click on the Buffer Analysis toolbar (fit to window).
- The red line clearly shows the VBV overflow.
- ***Now, how to fix this?***



But what does it mean?

- This means that for the stream concerned, if the standard values are used, then there will be a VBV overflow unless something is changed - which means that the decoding rate will not keep up with the input data rate and therefore frames will be dropped (starting from displayed frame 54)
- So, this needs to be fixed...

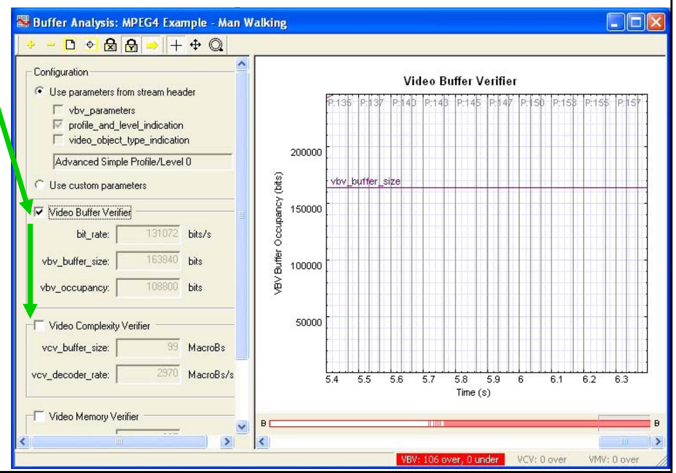
Fixing the VBV overflow - options

- There are a number of options
  - Increase the vbv_buffer_size so that it does not overflow.
  - Decrease the bit-rate, i.e. the rate at which the encoder is sending the data to the decoder.
  - Increase the displayed frame rate, so that the decoder removes bits more quickly.
  - Increase the number of bits per frame (e.g. by changing the quantizer or increasing the frame size).
  - Change the Profile/Level indication, so that it stays within limits (for example, change to ASP at Level 2).
  - Use custom values in the vbv_parameters in the sequence header.

This tutorial will examine the first & second of these

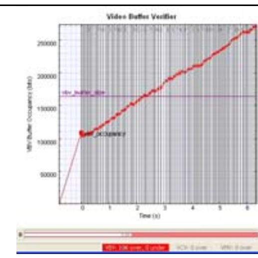
Fixing the VBV overflow

- Clear the check boxes for VCV and VMV so that only VBV remains (note: vertical scale is now bits, not %).
- *Note:* VBV graph has disappeared because it is off the scale set for the VBV.



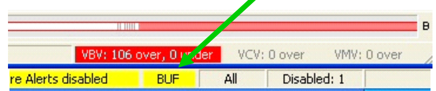
Fixing the VBV overflow (2)

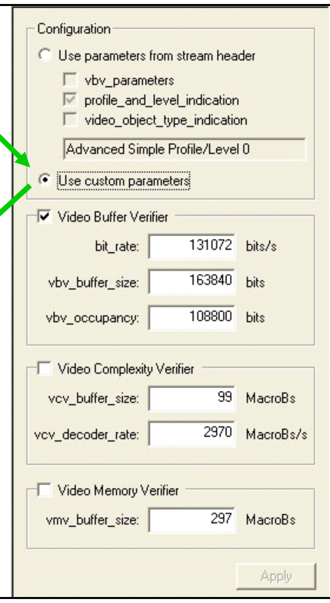
- So click on the Buffer Analysis toolbar again to see it all
- Zoom in to see the detail - click 3 times
  - the 'Y' scale of the graph is locked (shown by) - this keeps the 'Y' scale at a useful zoom level
  - the VBV graph is higher at the end of every frame than at the start - showing that the VBV buffer is being filled faster than the decoder can empty it
- Then zoom out again (press 3 times)



Fixing the VBV overflow (3)

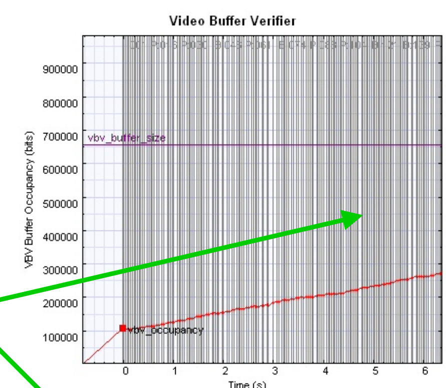
- Select 'Custom parameters'.
- ...The word 'BUF' appears in a yellow box in the main MTS4EA status bar, indicating that custom buffer parameters are in use.

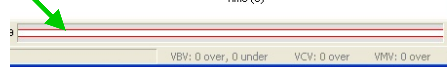




VBV overflow fix - increase buffer size

- The default vbv_buffer_size for ASP L0 is 163,840 bits = 20,480 bytes: not very large
- Try entering 655,360 bits = 80 kbytes (click 'Apply' then to fit to window)
- *SUCCESS! the graph stays below the limit and no over-flows or underflows*



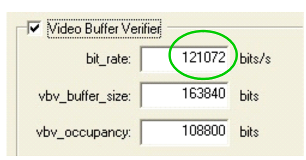


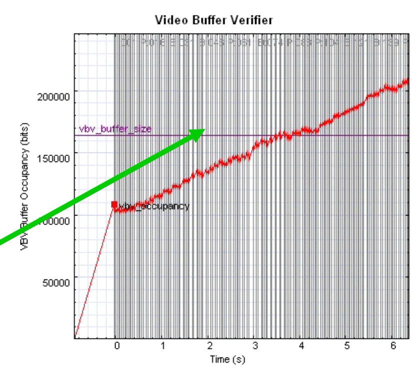
VBV overflow fix - reduce input data rate

- But what if your mobile device can't spare 80 kbytes of memory - only the ~20 kbytes (as given in the standard)?
- What does the input data rate have to be reduced to?
- And does this comply with the specifications of the network on which your device will operate?
- Reset the parameters by clicking 'Use parameters from stream header'; then reselect 'Use custom parameters'.

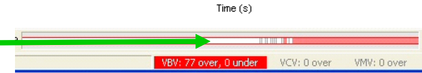
VBV overflow - reduce input rate (2)

- The VBV graph shows that the decoder is being supplied with data faster than it can decode - so reduce the bit rate (enter the value):
  - To 121072 bits (click 'Apply') and 'Fit to window' icon)



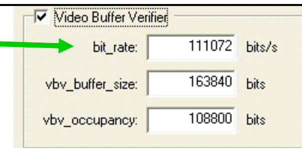


  - The graph still goes up (but not as much)
  - And there are fewer frames with overflow

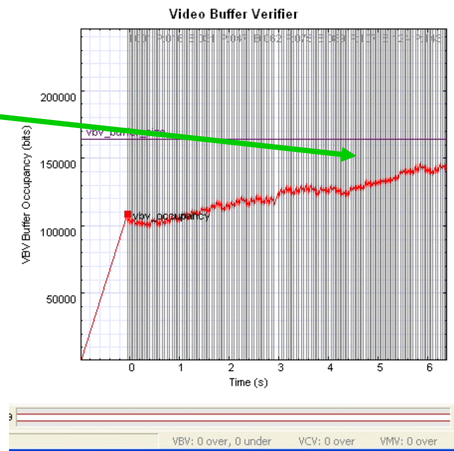


VBV overflow - reduce input rate (3)

- ... Then adjust to 111072 bits, click 'Apply' and 'Fit to window' icon ( )



- *SUCCESS! - the graph now stays below the buffer limit.*



> *<u>BUT</u> does the maximum rate of 111072 bits meet your network's specification?*

Other points to note; other features

- The values of the custom parameters (*if selected*):
  - Are used in the alert pop-ups and Trace file error reports.
  - When entered and 'Apply' is clicked, there is no need to replay the stream.

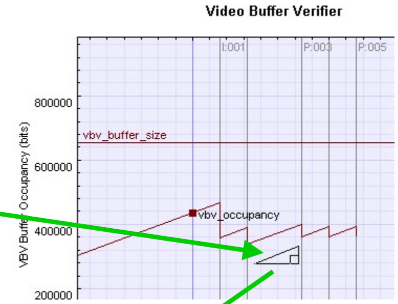
- The VBV display builds in real-time
  - Overflow is in **Red; ** underflow is in **Yellow.**
  - Each bar line fills and auto-scales.
    (corresponding with the horizontal scroll bar)
  - Each buffer item title on the status bar changes to **Red** or **Yellow** if there has been any overflow/underflow, and the number indicates in how many frames there is overflow/underflow.

Measuring values

- Use the ‘Measure’ tool + to measure the buffer occupancy at any point
  - the data is given in the status line (bottom left)

    Point: (0.266 s, 564093 bits)

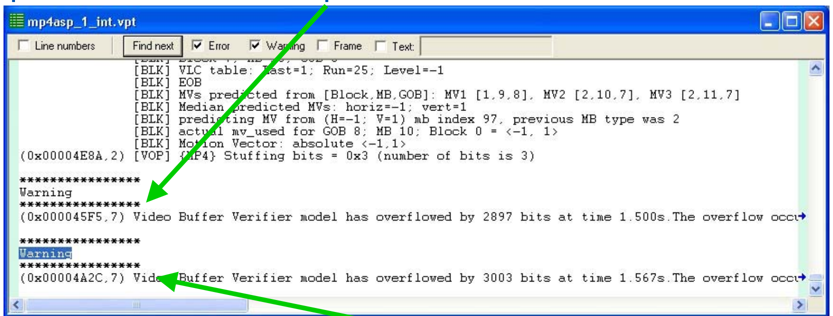
  - hold and drag the mouse to measure a gradient



Start: (0.181 s, 280397 bits) End: (0.312 s, 333178 bits) Delta: (0.132 s, 52781 bits) Gradient: 400445 bits/s

Buffer non-conformance reports in Trace files

- The buffer non-conformance also gives a pop-up alert and is reported in Trace / Interpret and Trace / Parse Bitstream files:



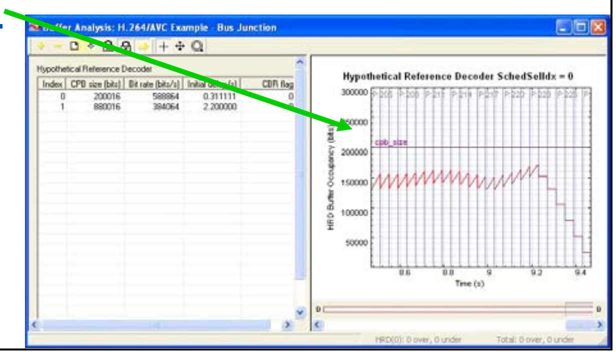
Note: (1) in the screenshot above there are **two** overflow reports together (in this case for frames 47 and 49), but which are listed *after* frame 50 in the Trace file - this is due to the decode ordering in the sequence

(2) the warnings generated in the Trace file *will vary with the values of the buffer parameters*, corresponding with alerts in the buffer analysis view

HRD buffer analysis in H.264/AVC

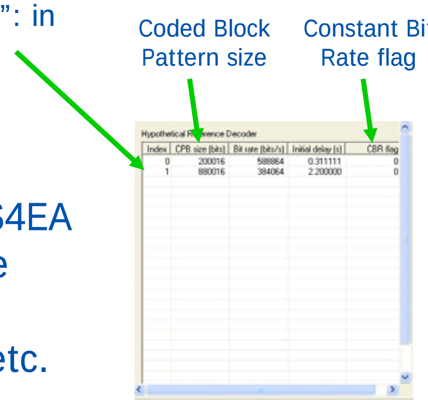
Note: not all H.264/AVC streams have HRD parameters in them (many do not): if the HRD parameters are not present then buffer analysis is disabled in MTS4EA.

- Load the H.264/AVC example file ‘Bus Junction’; click the buffer analysis icon ; and play to the end of the stream.
  - The HRD graph builds as the stream plays.



HRD buffer analysis in H.264/AVC (2)

- HRD analysis is far more complex than for MPEG-4 or MPEG-2:
  - HRD can have multiple “buckets”: in this case, two (indexes 0 and 1)



- Due to the complexity, MTS4EA does not currently allow the user to alter the values for CBP, bit rate, initial delay, etc.

Conclusions

- The Buffer analysis of the MPEG-4 example ‘Man Walking’ shows that this stream does not comply with the buffer constraints - and precisely in which frames the overflows occur.
- It also shows two possible fixes:
  - either exactly how much memory would need to be allocated.
    Can your hardware allow this much memory usage?
  - or the input data rate that should not be exceeded.
    Is this correct for your network?
- MTS4EA has shown *exactly* where the problems are and you have the information to fix the issue.

Tutorial 10: MPEG-2 Program Streams: Person Track and Grenadier Guards

This tutorial focuses on:

- Opening an MPEG-2 Program stream
- Syntax errors:
  - ‘Person Track’ : invalid VLC for ‘dct_differential’
  - ‘Grenadier Guards’: ‘frame_rate_extension’ denominator and numerator equal but not zero
- HexView
  - Going to/from the Alert pop-up
  - Examining the bitstream in the HexView
- Looking at the MPEG-2 Program stream structure
- Looking at MacroBlock types in interlaced video

It uses the supplied video clips in the 'File' menu: ‘Example files’, 'MPEG-2 Program streams', ‘Person Track’ and 'Grenadier Guards'.

Basic functions

	Forwards		Backwards	
	Icon	Ctrl+	Icon	Ctrl+
Play video		P		Shift+P
Stop video		S		S
Pause/step one frame		A		Shift+A
Fast forwards/backwards		F		Shift+F
'Blind' fast forwards/backwards (no video displayed)		B		Shift+B
Pause on frame (then Fast fwd/Blind fast fwd to get to this frame quickly)				
Skip to next frame type/number/time		K		Shift+K

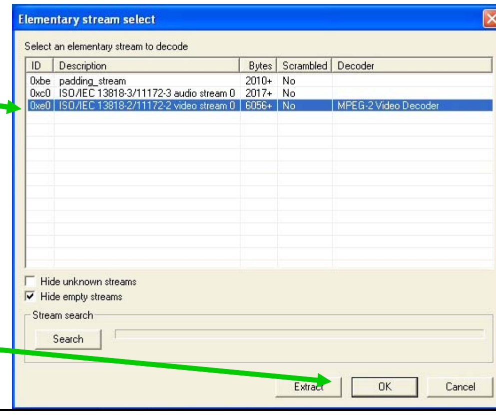
Tracks in an MPEG-2 Program stream

- Click 'File' menu, 'Example files', 'MPEG-2 Program streams', 'Person Track': the dialog box appears listing the tracks in the MPEG-2 program stream:

Video track, ID=0xe0 there can be multiple video tracks: the first video track is selected automatically.

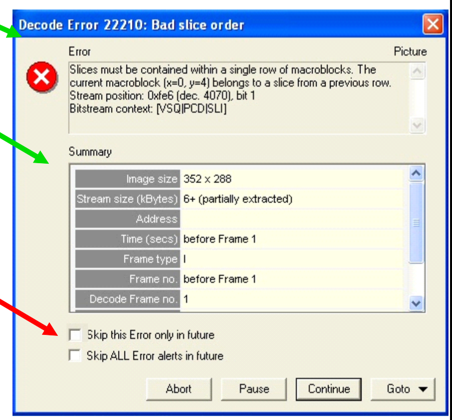
Other tracks with audio and 'meta' data (unsupported in MTS4EA).

Click 'OK'.



Syntax error (in Person Track)

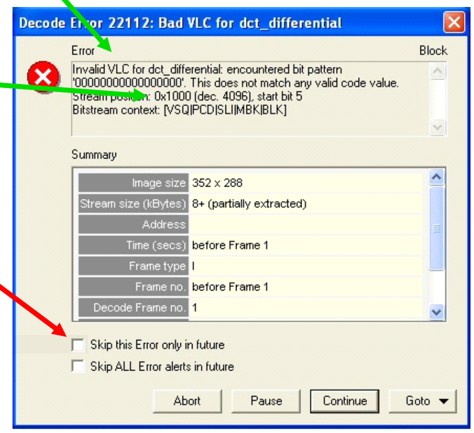
- Click 'Pause/Step forward', or the icon, once.
- An error pop-up occurs:
- This section of the alert gives a summary of the stream up to that point.
- Click 'Continue' (do *not* click 'Skip').



(See 'How to use MTS4EA' section of the user manual for information on Alerts)

Another alert…

- An invalid VLC has been used for the 'dct_differential' field.
  - ...at bitstream byte address 0x1000, bit 5:
- Click 'Continue' (do *not* click 'Skip'):
- ...and repeat this for the next error ('Bad slice order').

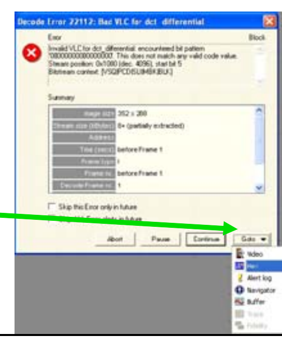


(See 'How to use MTS4EA' section of the user manual for information on Alerts)

The error can be seen in the video

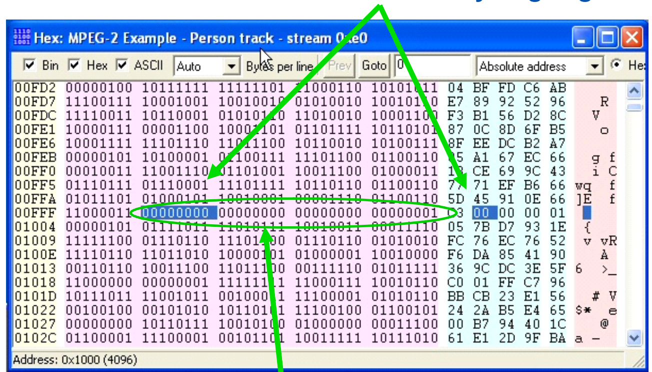
- The error is visible in frame 1:

- Now click stop and then 'Pause/Step forward' again
- 'Continue' past the *first* error, but go to the HexView of the bitstream *at the Invalid VLC error* by clicking 'Goto', 'Hex' (and 'Continue' past the next error).



VCL error area shown in HexView

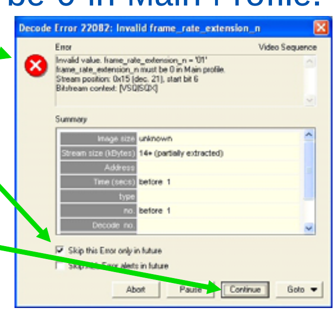
- Now look at the HexView:
  - The address of the error is automatically highlighted.



- Looking at the bits around the error - is this a start code emulation?

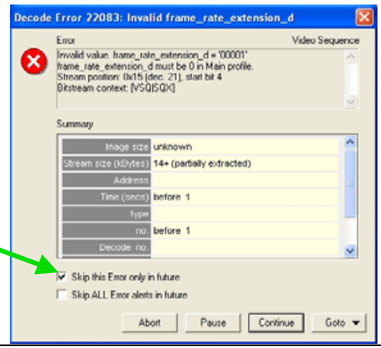
Alert in Grenadier Guards

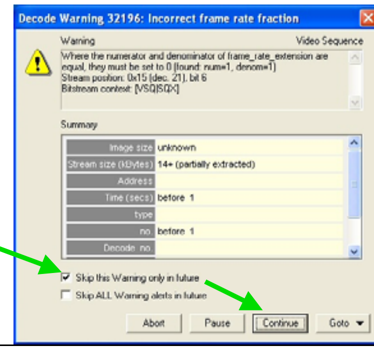
- Open the 'Grenadier Guards' MPEG-2 example stream.
- An error alert: the value of the denominator of 'frame_rate_extension' is 1 - but it must be 0 in Main Profile.
- Click 'Skip this Error only' (MTS4EA will still trigger on other warnings/errors).
- Click the 'Continue' button.
- The status bar indicates 1 disabled error.



More alerts …

- Two more pop-ups occur: an ‘error’ and a ‘warning’
  - the warning also is a response to the same area of error as the two errors - *this is a common occurrence: one bitstream error triggers multiple different alerts*
- Click ‘Skip this’ and ‘Continue’ for both errors - *do not play*





Look at the Program stream structure

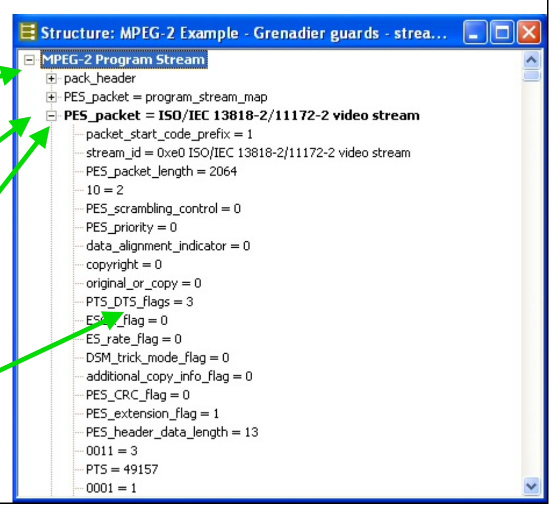
- Click on ‘Analysis’ , ‘View file structure…’ or click icon (or press Ctrl+R)

Top-level in the MPEG-2 Program stream

Video-related packs/PES in **bold**

Click to ‘**+**’ next to each item name to expand it (changes to ‘**-**’ when fully expanded)

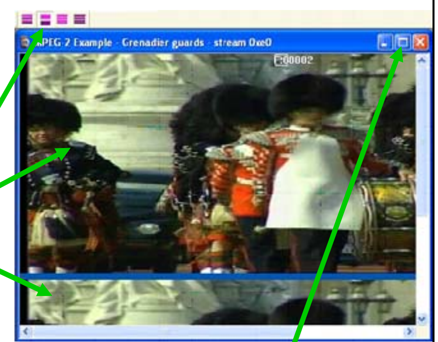
Data for the item



Look at the video - interlace view

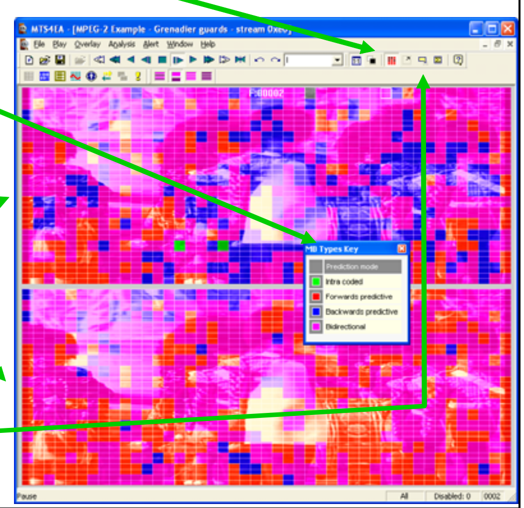
- The interlace toolbar appears, as the stream is interlaced (this toolbar can be moved).

- Click 'Pause/Step forward' (Ctrl+A or) to frame 2.
  - Click the Interlace toolbar 'Split fields' view.
  - The top and bottom fields are displayed separately.
  - Only part of the video window is shown - maximize it.



MacroBlock types in interlaced video

- Click the MB types overlay
- The MB types key can be dragged - (*hold <Ctrl> to force un-docking*)
- Different MB types can be seen in the the top and bottom fields.
- The MB tooltip can be used to get detailed information on each MB.



Conclusions

- There is one error in the 'Person Track' stream:
  - The error is frame 1, at bitstream location 0x1000 bit start 5.
  - This error affects the visual quality.
- There are two errors and one warning in the 'Grenadier Guards' stream:
  - The errors are in the header (before frame 1), at bitstream location 0x15 bit start 6.
  - The warning alert also highlights on the same error area.
  - This appears *not* to affect the visual quality.
- The different MacroBlock types in each of the top and bottom fields can be easily seen.

Tutorial 11: Fidelity Analysis

This tutorial focuses on:

- Fidelity analysis in real-time (PSNR etc.)
  - Using YUV source reference file
  - Fidelity metrics available
  - Icon controls for navigation through the graphs
- Comparing fidelity of the same bitstream compressed with MPEG-2 and H.264/AVC
- Viewing the visual difference
  - Between compressed bitstream and YUV source reference video

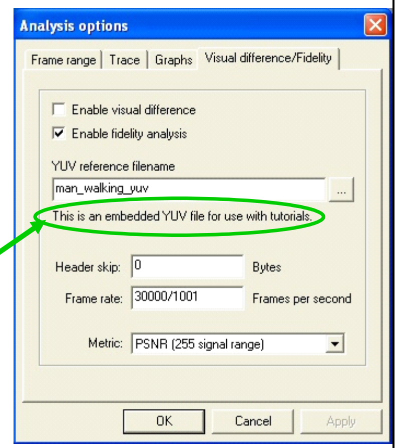
It uses the supplied video clips in the 'File' menu: ‘Example files’, 'MPEG-4 Elementary streams', 'Man Walking’; H.264/AVC 'Grenadier Guards' and MPEG-2 'Grenadier Guards'.

Prior tutorials assumed … (1 - 3, 9)

- It is assumed that you have done the previous tutorials and know how to:
  - Open example streams; play/stop/advance/rewind etc. the video.
  - Set the frame range.
  - Set the Trace file name and options; view the Trace files.
  - Use the alert pop-ups and interpret them.
  - Use the icons and controls on the buffer analysis graphical display - the icons are the same and the graphical display is very similar for the fidelity analysis.
- It is also assumed that you *already* understand the principles of fidelity analysis (MTS4EA does *not* teach this, although the user manual explains the fidelity metrics available).

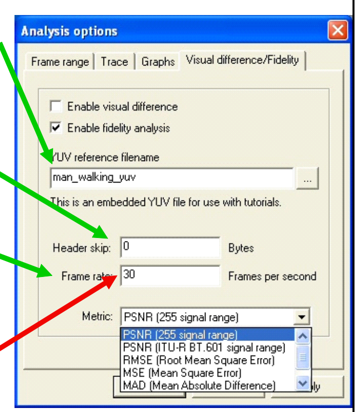
Begin fidelity analysis …

- Open the Example MPEG-4 Elementary stream ‘Man Walking’ (do not play it yet).
- Click ‘Analysis’ menu, ‘Fidelity enable…’
  - The dialog box appears for the YUV reference file which will be used to compare with the compressed video file
  - In this case, the YUV reference file is provided within MTS4EA as an example file (as indicated by the text under the filename box)



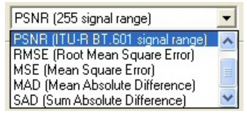
YUV reference file

- The YUV reference file is often the original source video file which was used when the compressed video was encoded. Comparing the compressed video with the original source quantifies the amount of artifact introduced by the encoder.
- Number of header bytes to skip past at the start of the YUV file.
- Frame rate of the YUV file:
  - This MUST be correct for MTS4EA to be able to associate the correct YUV frame with the corresponding compressed frame.
  - **set this to ‘30’.**



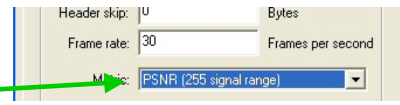
Fidelity metrics available

- Six different fidelity metrics are available:
  (see the user manual for a full definition of the metrics)



  - Peak Signal to Noise Ratio (with signal range 0-255)
  - Peak Signal to Noise Ratio (with the ITU-R BT.601 signal range, which is e.g. maximum value of 220 in the Y plane)
  - Root Mean Square Error (the square root of Mean Square Error)
  - Mean Square Error
  - Mean Absolute Difference
  - Sum Absolute Difference

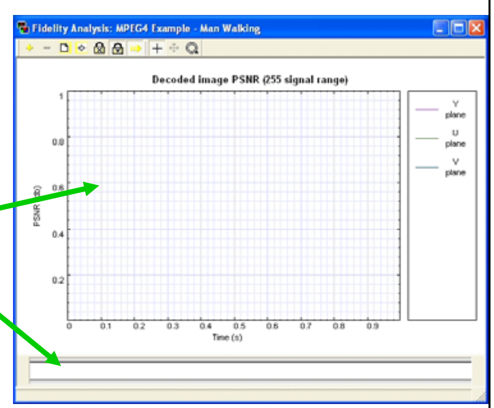
- Select 'PSNR (signal range 255)' and click 'OK'



Opening the fidelity analysis view

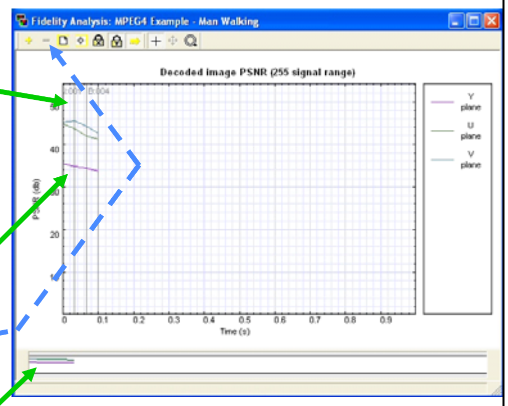
- The video window is then shown, but the fidelity analysis icon is now enabled in the 'Views' toolbar:

- Click the icon and the fidelity analysis graph view appears:
  - The graph is empty so far, because no video has been played.

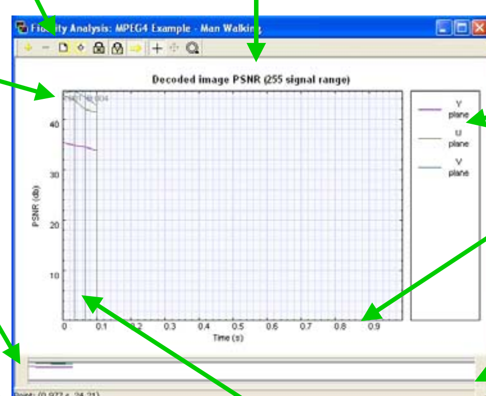


Opening the fidelity analysis view (2)

- Click the pause/step forward icon 4 times (or Ctrl+A).
- Note:
  - The graph area is *not* filled in with **every** click. This is because this bitstream includes B-VOPS and a number of frames must be decoded before each frame can be displayed.
  - All the YUV PSNR values may not be visible. This is because the graph auto-scales with the first few values. **Click this icon** to rescale the graph.
  - The thumbnail bar view shows that the fidelity data has been collected.



The fidelity analysis window

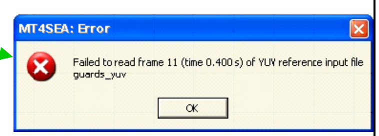


Fidelity analysis - icon controls

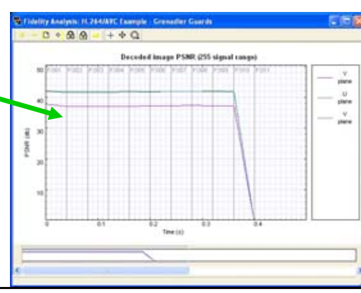
Icon	Function
	Zoom in (+) and zoom out (-) centered on the center of the window - affected by the 'Lock X/Y' icons (see below)
	Fit all data into window
	Goto origin (zero), i.e. start of video sequence
	Lock X/Y zoom in/out and scrolling/panning. E.g. when the 'Lock Y' button is pressed, zoom in and 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, i.e. fill the analysis data in real-time, as the video is being decoded and scroll the window to the right
	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) - this is affected by the 'Lock X/Y' icon - see above
	Zoom in/zoom out centered on the location of this cursor - this is affected by the 'Lock X/Y' icon - see above. Press the <shift> key to zoom out

Comparing H.264/AVC with MPEG-2

- Load the H.264/AVC example file 'Grenadier Guards':
  - *Make sure* to set the YUV frame rate to **25**
  - Click 'Skip all errors' when the alert pops up in frame 8
  - There is an error in frame 11: this is because there is only 10 frames of YUV data (this is OK)

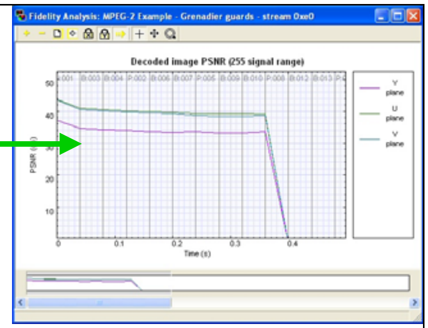


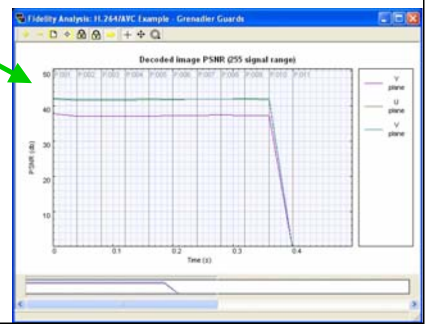
  - Graph result:



Comparing H.264/AVC with MPEG-2 (2)

- Repeat with the MPEG-2 Program Stream example 'Grenadier Guards' (the same sequence):
- Compare with the H.264/AVC PSNR
- The PSNR is higher (better) for H.264/AVC in Y, U and V

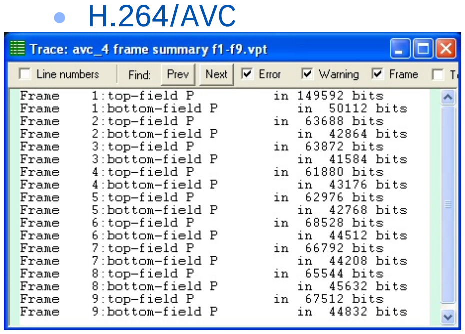




Comparing H.264/AVC with MPEG-2 (3)

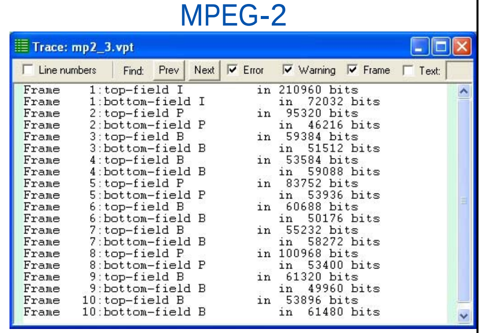
- Now look at number of bits used in frames 1-9 of the H.264/AVC and MPEG-2 'Grenadier Guards' sequences:
  - H.264/AVC

MPEG-2



Trace: avc_4 frame summary f1-f9.vpt

```
Frame    1:top-field P        in 149592 bits
Frame    1:bottom-field P       in  50112 bits
Frame    2:top-field P        in  63688 bits
Frame    2:bottom-field P       in  42864 bits
Frame    3:top-field P        in  63872 bits
Frame    3:bottom-field P       in  41584 bits
Frame    4:top-field P        in  61880 bits
Frame    4:bottom-field P       in  43176 bits
Frame    5:top-field P        in  62976 bits
Frame    5:bottom-field P       in  42768 bits
Frame    6:top-field P        in  68528 bits
Frame    6:bottom-field P       in  44512 bits
Frame    7:top-field P        in  66792 bits
Frame    7:bottom-field P       in  44208 bits
Frame    8:top-field P        in  65544 bits
Frame    8:bottom-field P       in  45632 bits
Frame    9:top-field P        in  67512 bits
Frame    9:bottom-field P       in  44832 bits
```



Trace: mp2_3.vpt

```
Frame    1:top-field I        in 210960 bits
Frame    1:bottom-field I       in  72032 bits
Frame    2:top-field P        in  95320 bits
Frame    2:bottom-field P       in  46216 bits
Frame    3:top-field B        in  59384 bits
Frame    3:bottom-field B       in  51512 bits
Frame    4:top-field B        in  53584 bits
Frame    4:bottom-field B       in  59088 bits
Frame    5:top-field P        in  83752 bits
Frame    5:bottom-field P       in  53936 bits
Frame    6:top-field B        in  60688 bits
Frame    6:bottom-field B       in  50176 bits
Frame    7:top-field B        in  55232 bits
Frame    7:bottom-field B       in  58272 bits
Frame    8:top-field P        in 100968 bits
Frame    8:bottom-field P       in  53400 bits
Frame    9:top-field B        in  61320 bits
Frame    9:bottom-field B       in  49960 bits
Frame   10:top-field B        in  53896 bits
Frame   10:bottom-field B       in  61480 bits
```

  - *The MPEG-2 sequence also uses ~20% more bits*

Comparing H.264/AVC with MPEG-2 (4)

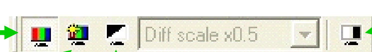
➢ More areas to consider/investigate:
- What encoder parameters were used for H.264/AVC and MPEG-2 (what frame types, etc.).
- Look at the areas of lower PSNR - can either the H.264/AVC or the MPEG-2 encoding parameters be changed to improve these lower areas without unduly affecting the good areas?
- Compare other fidelity analyzes also - do they show the same differences?
- *And just because the PSNR is better for the H.264/AVC sequence, it does not mean it looks better* - look also at the visual differences...

Visual difference: encoded <-> original

➢ Now see if the visual differences between the encoded stream and the original are more noticeable for MPEG-2 or H.264/AVC:
- Load the MPEG-2 stream 'Grenadier Guards' again.
- From the 'Overlay' menu, select 'Visual difference' then 'Enable'; set the frame rate to 25; click 'OK'.
- The Visual difference toolbar appears:

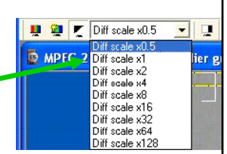
View luma only (no U, V)

View encoded



View YUV reference

View difference: when selected, drop-down menu enabled (this expands the visual difference)

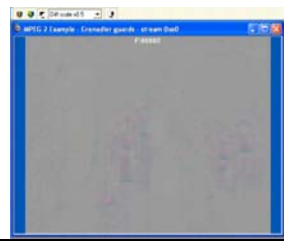


Visual difference (2)

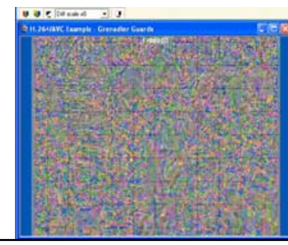
- View encoded

- View original

- View difference (x0.5)



- View difference (x8)



Visual difference (3)

- Now repeat with H.264/AVC 'Grenadier Guards':
  - or *better still, open a second MTS4EA window* and look at the MPEG-2 and H.264/AVC on-screen at the same time.
- Can the differences between the visual quality of the MPEG-2 and the H.264/AVC be seen?

The visual quality analysis can be done in Batch mode - enabling the automated testing of many different encoding options to check the effect on visual quality and thereby optimize the quality.

Conclusions

- MTS4EA provides the means to begin assessing visual quality by:
  - Using metrics such as PSNR.
  - Doing a visual difference between the encoded video and the reference (source) video used for the encoding.
- Visual quality analysis and visual quality improvement require that many items are considered, such as the choices made of the encoding parameters (frame types, motion vectors, etc.) and bit-rates of encoding.
- Visual quality analysis is an involved process - MTS4EA provides the tools to start doing this in a quantifiable and logical fashion.

.

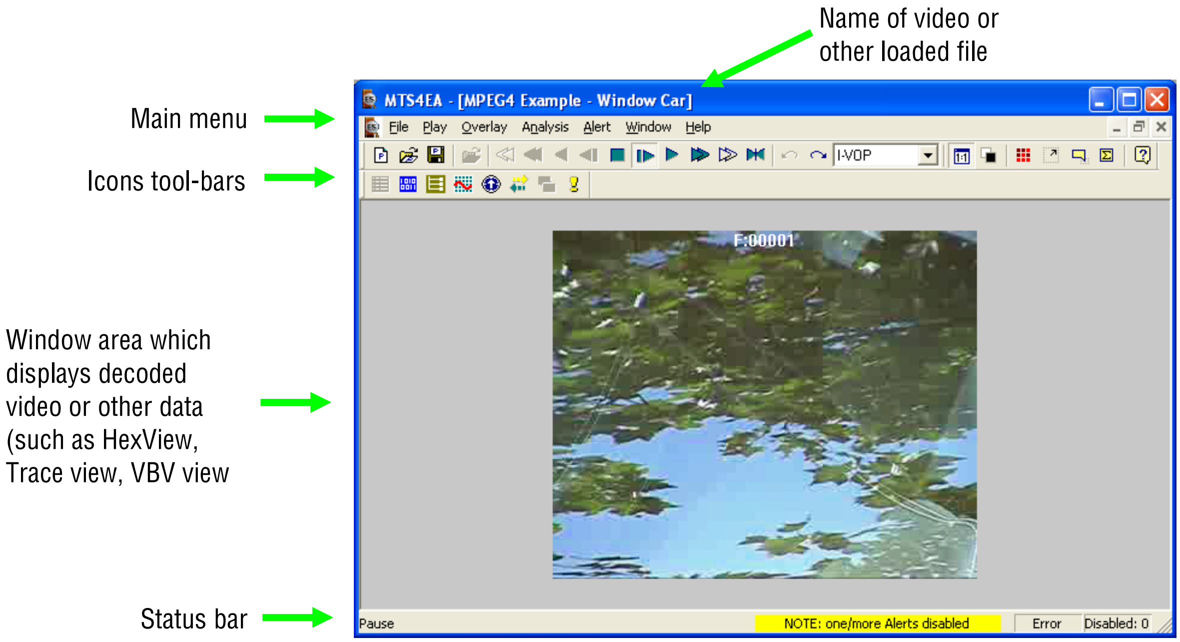
How to Use the MTS4EA

How to Use the MTS4EA

This chapter:

- Describes the elements of the MTS4EA display window
(see *Window Elements*, page 7-2)
- Describes how to start using MTS4EA display
(see *Starting to Use MTS4EA*, page 7-3)
- Describes each of the menu items in detail and how to use them
(see *Main Menu*, page 7-11 to *Help Menu*, page 7-194)
- Explains the icon toolbars, below the main menu
(see *Icon Toolbars*, page 7-196)
- Explains the context-sensitive toolbars, which appear depending upon the video standard used and/or the actions being performed
(see *Context-sensitive Toolbars/Tooltips*, page 7-201)
- Explains the information provided on the status bar
(see *Status Bar*, page 7-206)
- Lists the shortcut keys, such as ‘Ctrl+A’ to Pause/Frame Advance one frame
(see *'Ctrl' Shortcut Keys*, page 7-208)
- Explains use of the ‘Alt’ key to select menu items
(see *‘Alt’ Menu Keys*, page 7-210)
- Explains use of the command line/batch mode
(see *Command Line/Batch Mode*, page 7-210)

Window Elements

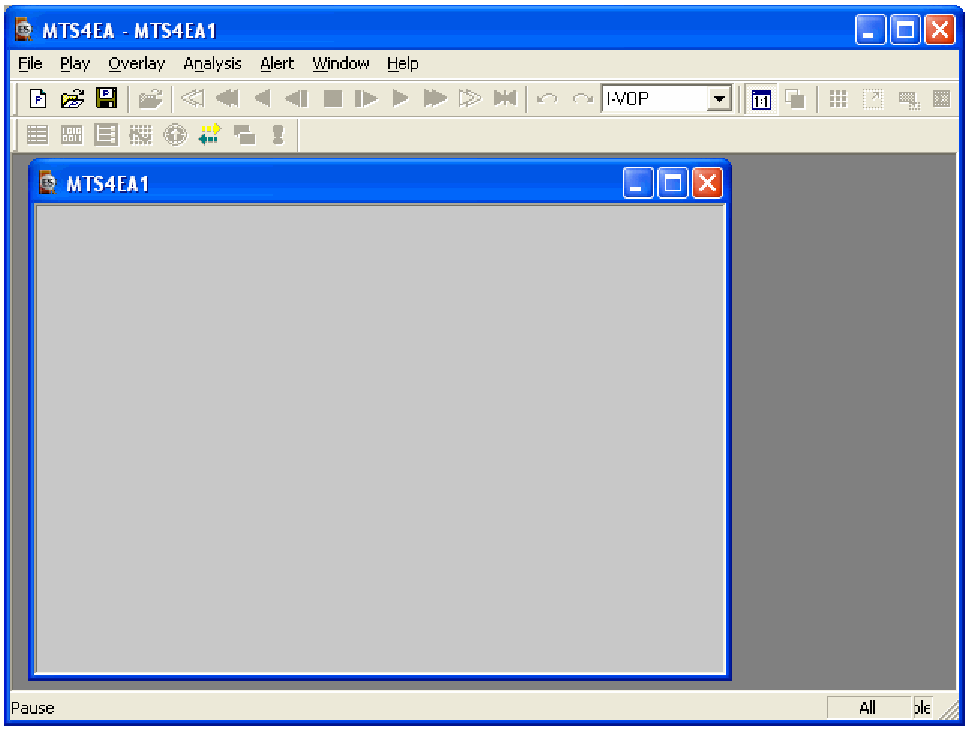


NOTE. *The above screen shot was taken with MTS4EA running under Windows XP. When MTS4EA is running under Windows 2000 there may be some visual differences in the borders, menus and the way that pop-up menus appear from those shown in the screen shots. However, there are no functional differences in the way MTS4EA operates on these different versions of Windows.*

NOTE. *In addition to the above toolbars/windows, there are multiple other toolbars which are visible when particular functions are enabled. Also, there are some toolbars, tooltips and menu items which are context-sensitive, i.e. 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 like the following screenshot:

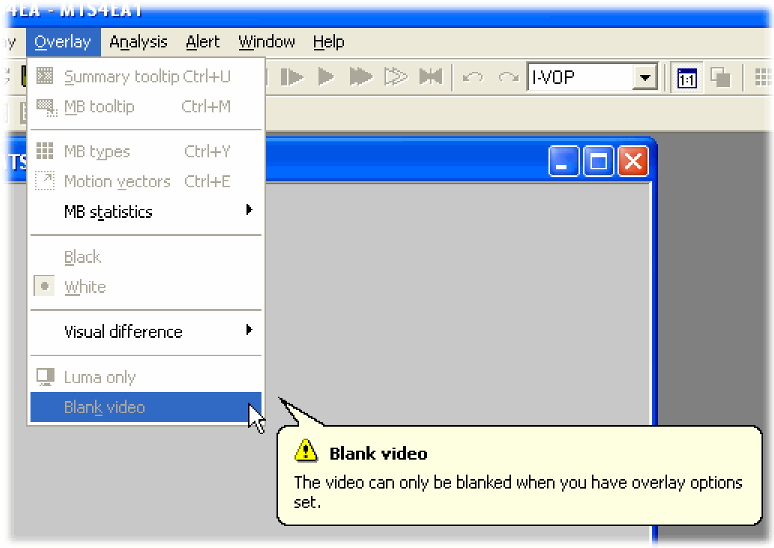


The gray background in the window indicates that no video or other files are loaded. The only icons on the toolbar which are currently enabled are the icons:

- for the Project ('New Project', 'Open Project' and 'Save Project')
- 'Open video...' 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, page 4-9* *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:

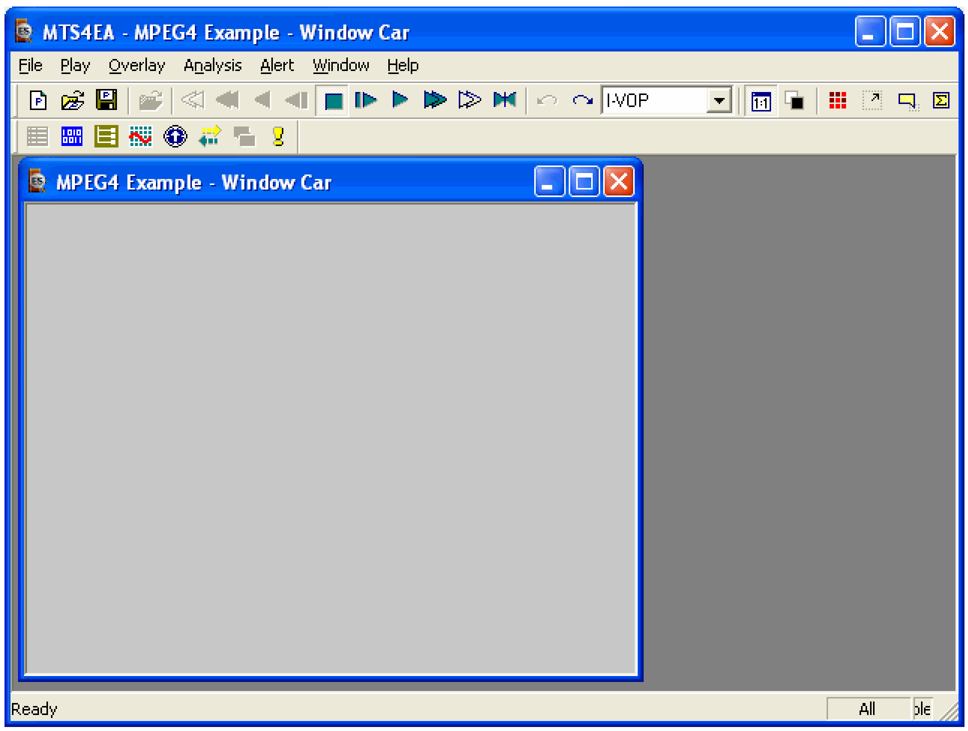


Opening a video file

To open a video file:

- click on 'File' - 'Open video...', or
- click the open video file icon , or
- press Ctrl+O

Once a video file is opened the display initially looks like the following screenshot:



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

If another video file is opened then this replaces the video file last opened - another video window is not opened.

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

Video playing mode: restrictions

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

- '**Stop**': video is stopped
- '**Play**': video is paused (after Pause/Step forward) or playing at normal speed or 'Fast forward' or 'Blind fast forward'

Various actions can only be performed or are only effective when the video is in the relevant modes. The manual states below where this is the case, in the relevant section. A few such examples are:

- the 'Trace enable' and 'Graph enable' options on the 'Analysis' menu are only available in '**Play**' mode (see *Analysis Menu*, page 7-101).
- the real-time overlays, such as 'MB types' and 'motion vector overlay' (see Overlay Menu, page 7-53) do not update the video window when in '**Stop**' mode

MTS4EA Projects

When working with MTS4EA, the 'projects' feature allows all current setups to be saved at any point and reloaded - for example, to allow work to recommence at exactly the same point. The project file includes information on the name of the video file being analyzed, the views open and the location of these, plus many other items. See File Menu, page 7-13 for more information.

Synchronized views/navigating the views

MTS4EA has many views of the video and other data:

- Video view
- HexView
- Trace views
- Buffer analysis view
- Video navigator view
- Fidelity view (PSNR etc. analysis)
- File structure view
- Excel graphs.

The first six of the MTS4EA views are linked. This means that right-clicking in one view will take you to the corresponding location in another view.

Example of 'Goto view' menus:

from Trace file

from video view

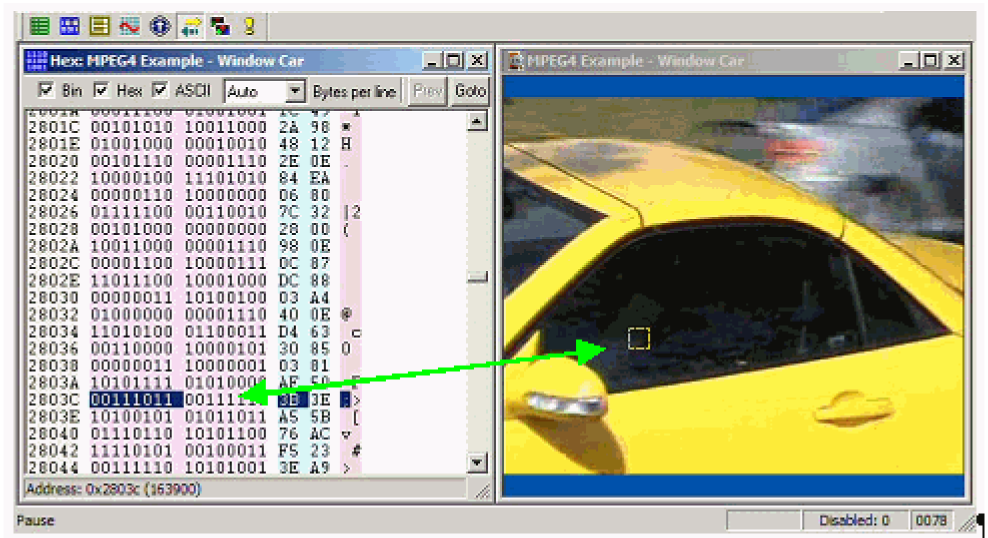
HINT. *When in the video view, to make it easier to see which area of the video window is being selected when linking to another view, there is always a small square or rectangle on the video view which highlights the MacroBlock which will be used for cross-referencing when another view is selected.*

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

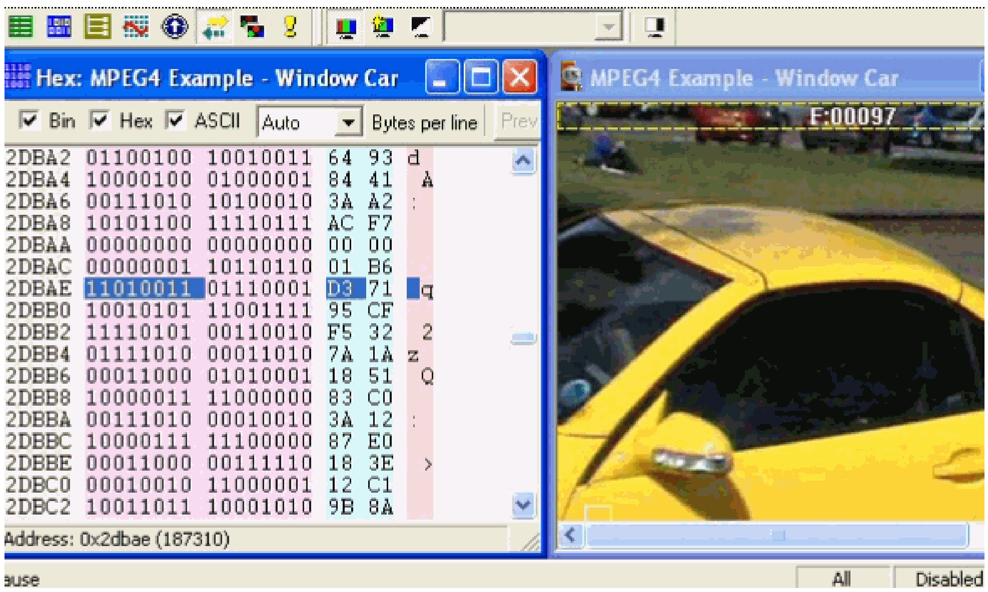
Likewise, selecting a specific byte in the HexView, then using the right-click context menu to select the 'Video' view will take the video window to the MacroBlock which contains the selected byte, and the MacroBlock is highlighted (see following screenshot).

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 (see below: the corresponding areas are pointed to the by green arrow);



- 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 (see below);



- if the selected area is in a frame or file header, i.e. 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 above: 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.

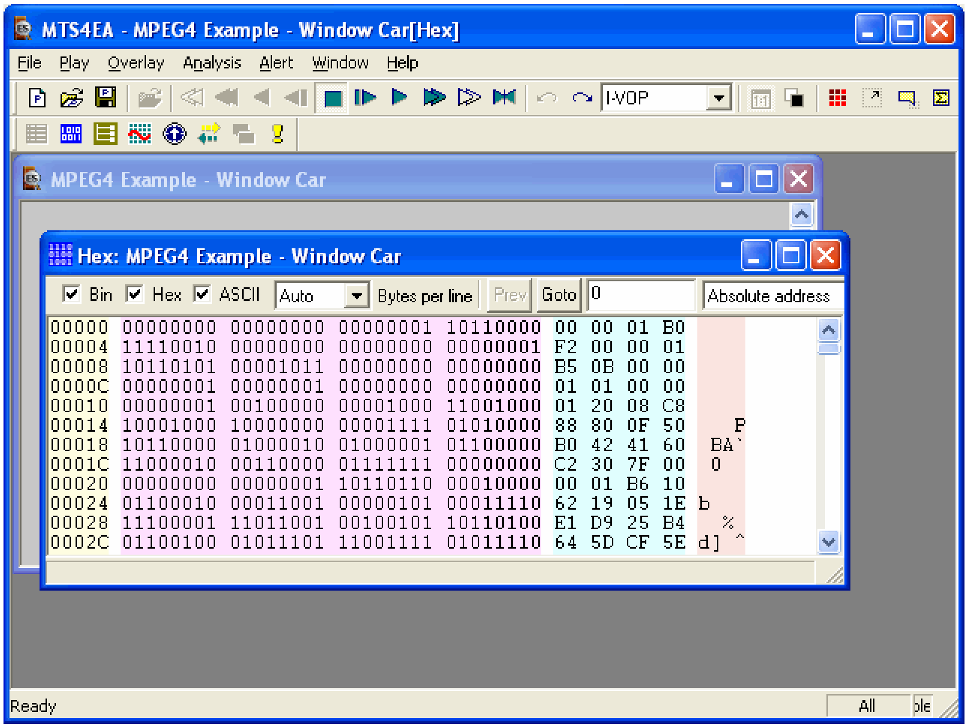
For example, if the video and HexViews are both open and the icon is pushed in and the video is then played, the highlighting in the HexView window advances so that the highlighted area always corresponds with the relative area in the video window.

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 below);



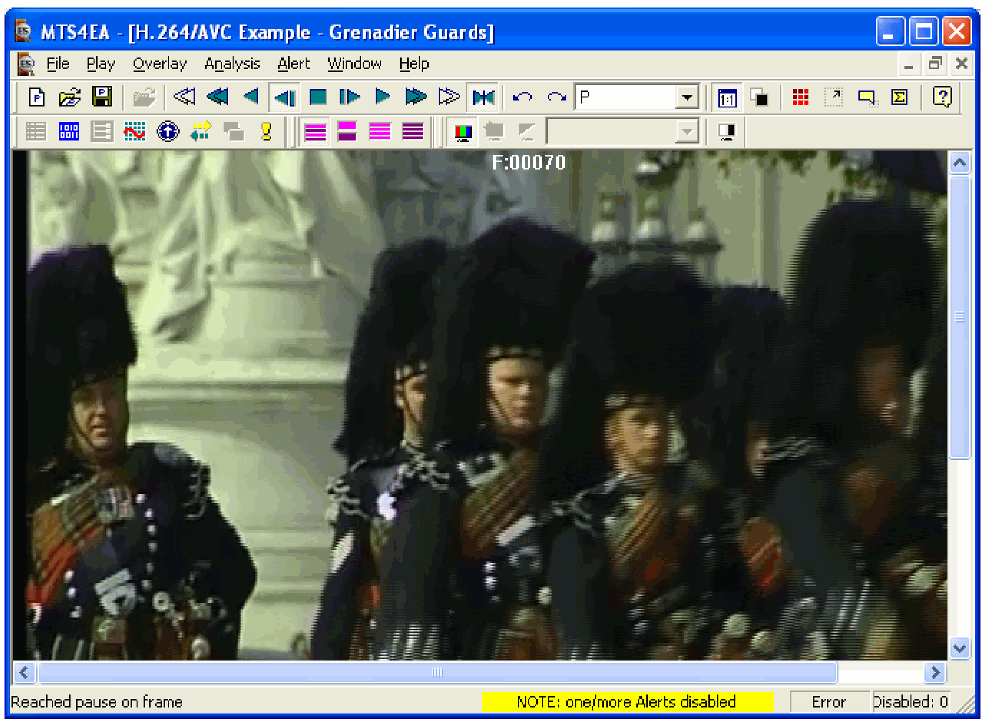
- other combinations, such as the video and HexView with the Trace file view and file structure view.

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

- for Trace views:
  - Select 'File', 'Open other...' (or 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), select 'Analysis', 'View graphs...';
- for HexView, select 'Analysis', 'View stream hex...' (or the icon or ‘Ctrl+H’;

- for the stream structure view, select 'Analysis', 'View file structure...' (or the icon or ‘Ctrl+R’);
- for the buffer analysis view, select 'Analysis', 'View buffer analysis...' (or the icon);
- for the fidelity analysis, select 'Analysis', 'View fidelity analysis...' (or the icon);
- for the Alert Log, select 'Alert', 'View alert log...' (or the icon).

Main Menu



The sub-menu options available underneath each main menu selection such as ‘File’, ‘Play’, 'Overlay', etc. are described in detail in the following sections. A summary of each main menu item is given in *Main Menu, page 7-11*.

Several of the common functions can be accessed via:

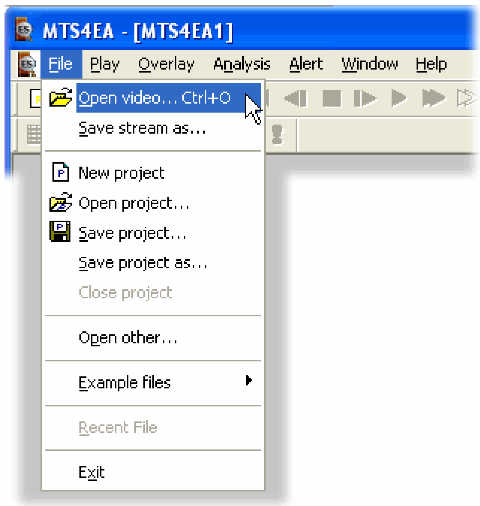
- the icon toolbars: see Icon Toolbars, page 7-196
- 'Ctrl' keys: see 'Ctrl' Shortcut Keys, page 7-208

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
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



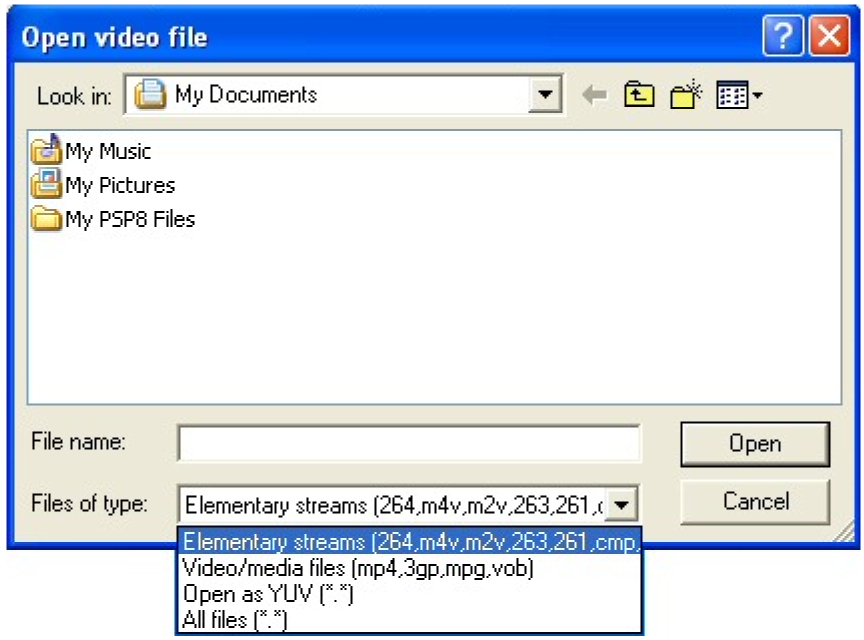
Open video...  Ctrl+O

With this option you can browse the file system to find a video file to analyze.

This can be:

- A 'container' file which includes other data such as audio and transport information as well as video, e.g. an MP4, 3GPP, MPEG-2 Program Stream or VOB file (see Permitted Video File Types/Formats, page 5-8),
- A compressed video file which only contains video data (such as an MPEG-4 video 'Elementary Stream', (see Permitted Video File Types/Formats, page 5-8),
- A YUV format file, (see Opening a YUV video file (any file extension), page 7-24).

The file dialog box shows a list of the different file types and the extensions assumed:



The default file extensions that MTS4EA looks for are:

- for video-only files: `.264`, `.m4v`, `.m2v`, `.263`, `.261`, `.cmp`, `.bits`
- for 'container' files including video: `.mp4`, `.3gp`, `.mpg`, `.vob`

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

NOTE. *The exceptions to this are VOB files, where MTS4EA will take different actions due to the VOB extension (see Open video... Ctrl+O, page 7-13) and 'Open as YUV' where MTS4EA will open as a YUV format file, irrespective of the file extension or the data found in the file.*

In addition:

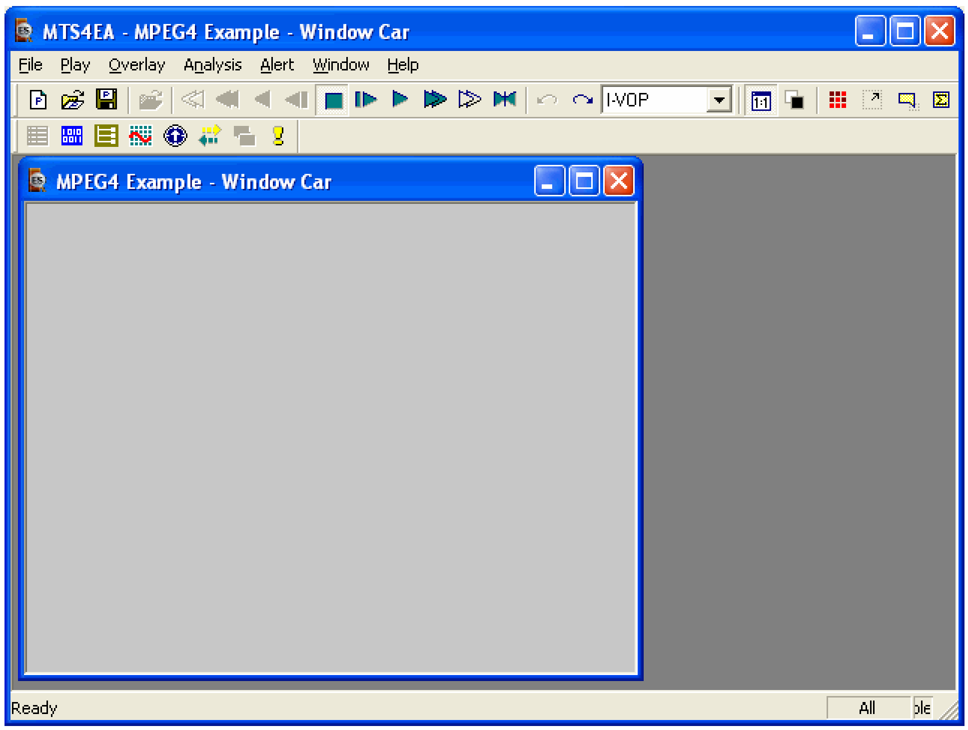
- 'Open as YUV' opens a video file assuming it is YUV format (irrespective of the file extension);
- 'All files' can also be selected: there is no restriction on the file extensions of video files that MTS4EA will open.

The selected file extension is reused by MTS4EA the next time a video file is opened.

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', i.e. one of the example files provided).

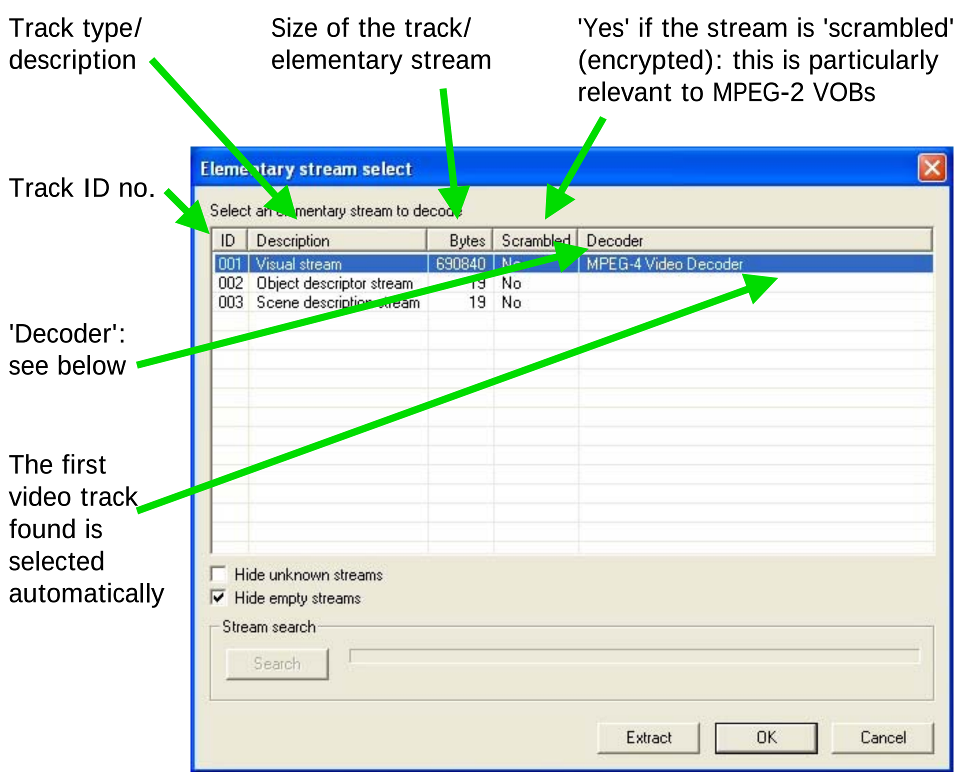
Opening a video 'container' file ('.mp4', '.3gp', '.mpg' etc.).

NOTE. *For the special additional dialog box for 'VOB' files; see Open video… Ctrl+O, page 7-13.*

Once the video container file is selected, the user is able to:

- play and analyze the selected video stream immediately within MTS4EA, or
- extract and save the selected video stream to a file before going on to play/analyze it.

With a 'container' video file, after clicking the 'Open' button on the 'Open video...' dialog box, MTS4EA will display a window as below (some of the controls are explained by arrows in the picture immediately below, others are explained with more text below this):



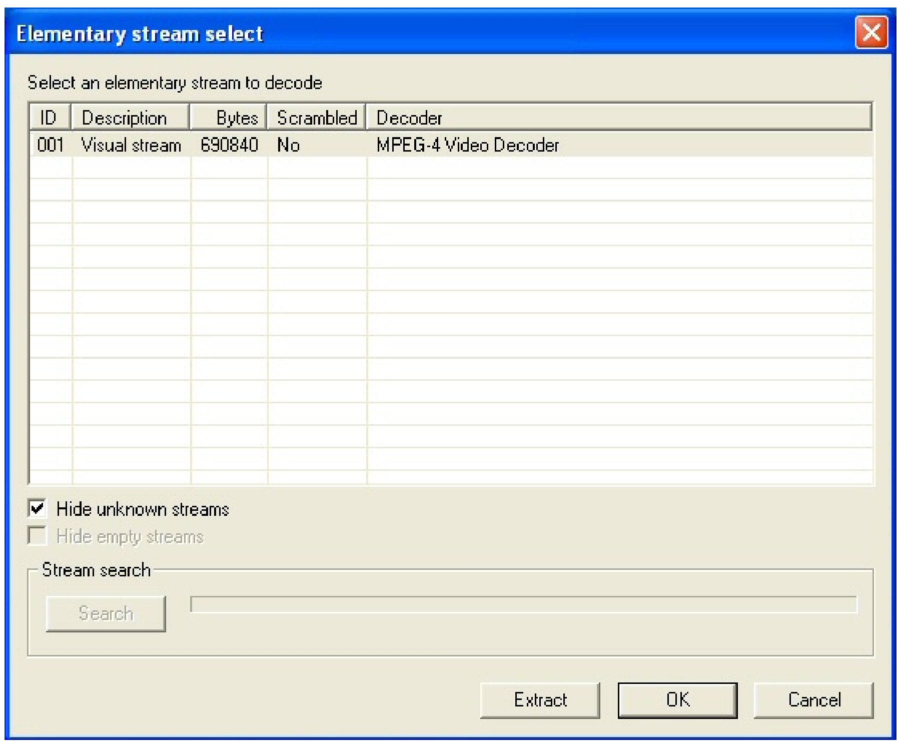
Decoder

The 'Decoder' column shows the video decoder which has been found within the stream concerned: there can be multiple streams containing video, each with different decoders.

If the 'Decoder' column is empty, then MTS4EA has been unable to match one of its decoders with the data found.

'Hide unknown streams'

When selected, 'Hide unknown streams' means that a stream is not shown if the MTS4EA has been unable to match one of its decoders. The example below shows the same MP4 container file as shown in the previous screenshot, but with the 'unknown' streams hidden:

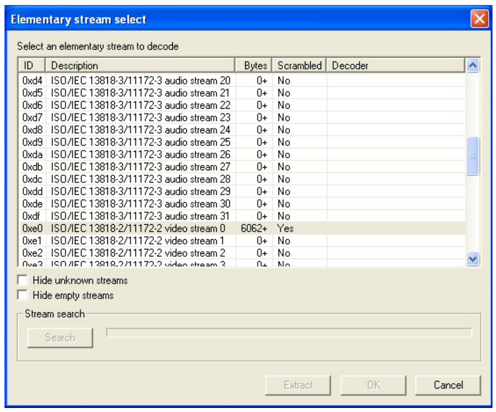


Note that when 'Hide unknown streams' is enabled, the empty streams are also hidden, because they do not contain valid data for which MTS4EA can select a decoder.

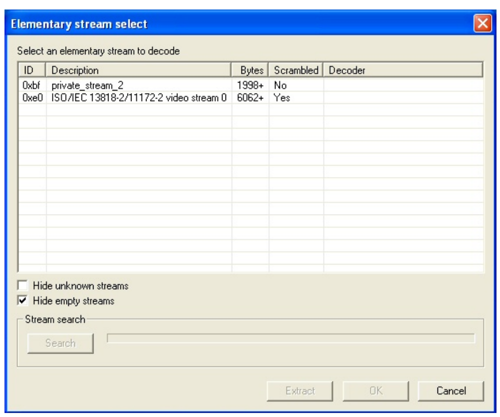
<u>'Hide empty streams'</u>

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 below)



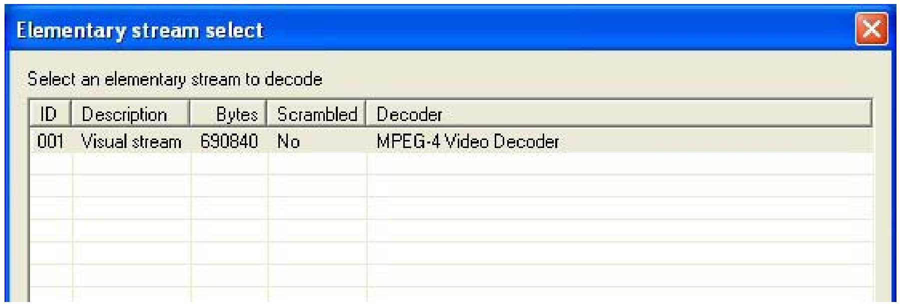
Empty streams shown



Empty streams hidden

'Search' (stream search)

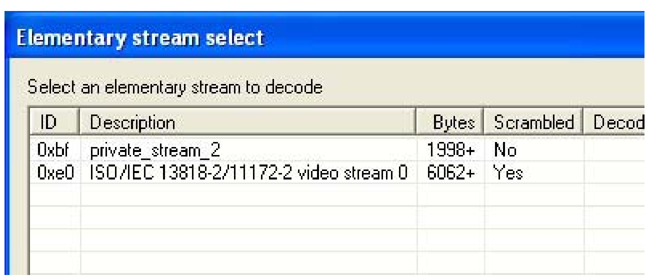
For some video standards, the total size of the video stream is indicated in the container file - for example, MPEG-4. In this case, the size of the elementary stream is as shown in the following screenshot:



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.

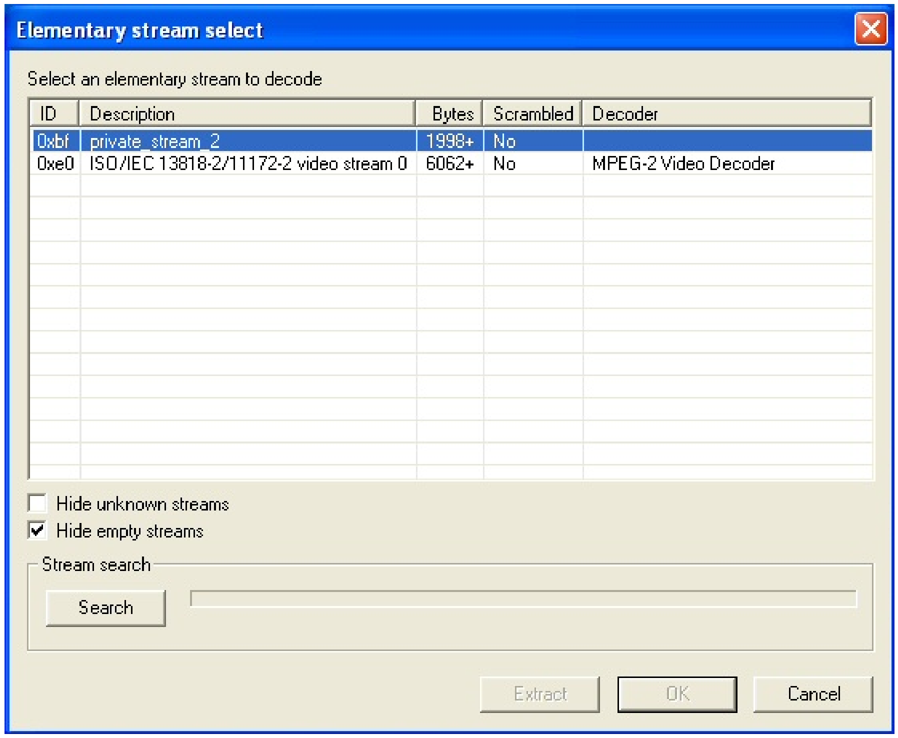
In this case, MTS4EA will display the size in the 'Bytes' column with a '+' sign to the right of the size, as shown below:



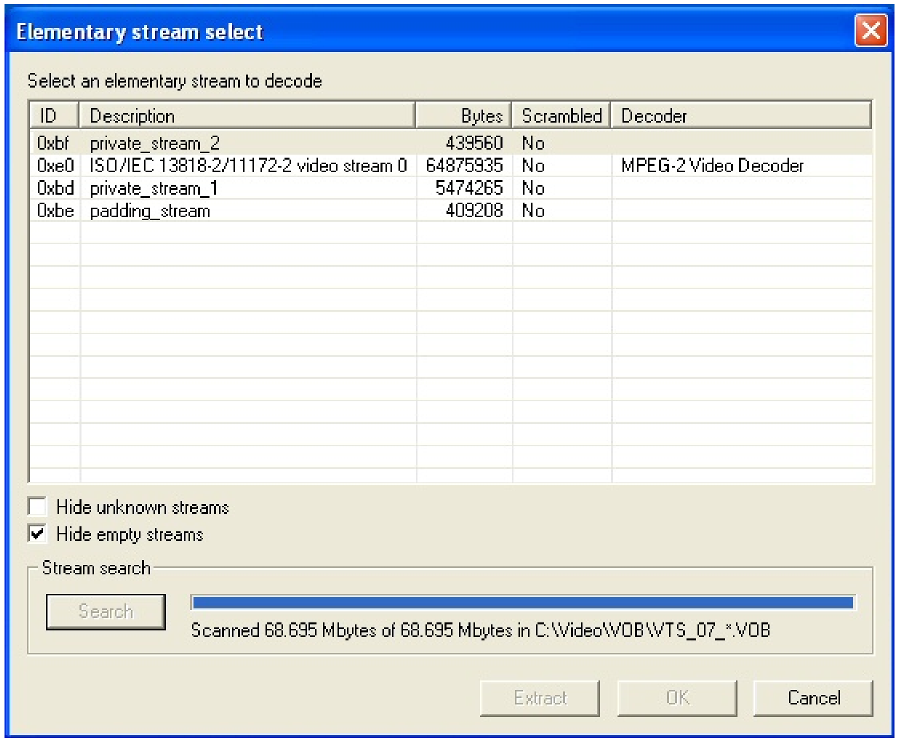
The '+' sign indicates that the whole file has not been searched and that there may be additional data.

***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, page 7-142.)

In order to find the total size of the whole video file, and to search other streams, use the 'Search' button. In the example below, the stream ID 0xbf, 'private_stream_2' is highlighted (i.e. selected), so if 'Search' is clicked, then this stream is searched.



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

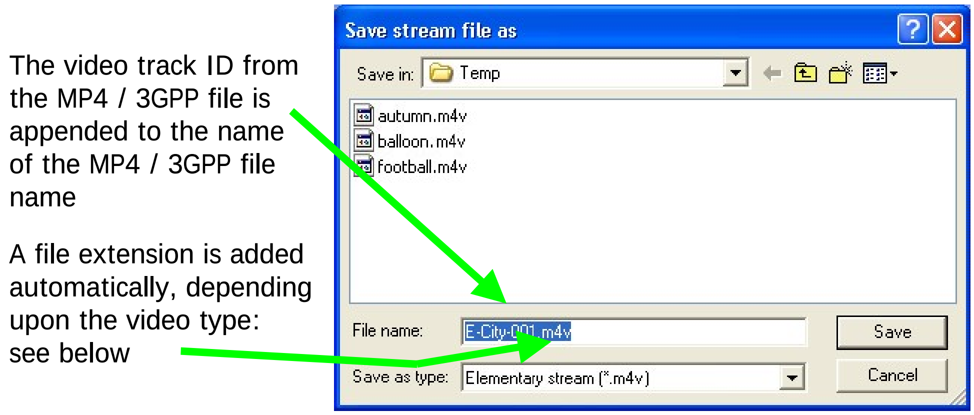


During the search, the blue progress bar fills, until the search is complete.

Note that during the search process, MTS4EA searches through all the linked files, and therefore may find other streams. It will be able to determine the total size of the streams found. This is why, in the previous screenshot, two additional streams are shown and the sizes no longer have '+' signs by them (the sizes are now known exactly).

<u>'Extract'</u>

This allows the highlighted video track to be saved as a separate video elementary stream:



NOTE. *If you decide to play and analyze the selected video stream immediately, then decide to* ***save it later****, this can be done by clicking the 'Save as...' option on the 'File' menu (See File Menu, page 7-13).*

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

- H.264/AVC video format files are given the extension: `.264`
- MPEG-4 video format files are given the extension: `.m4v`
  (except for MPEG-4 Short Header which is given the extension `.h263)`
- 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: `.h263`
- H.261 files are given the extension: `.261`

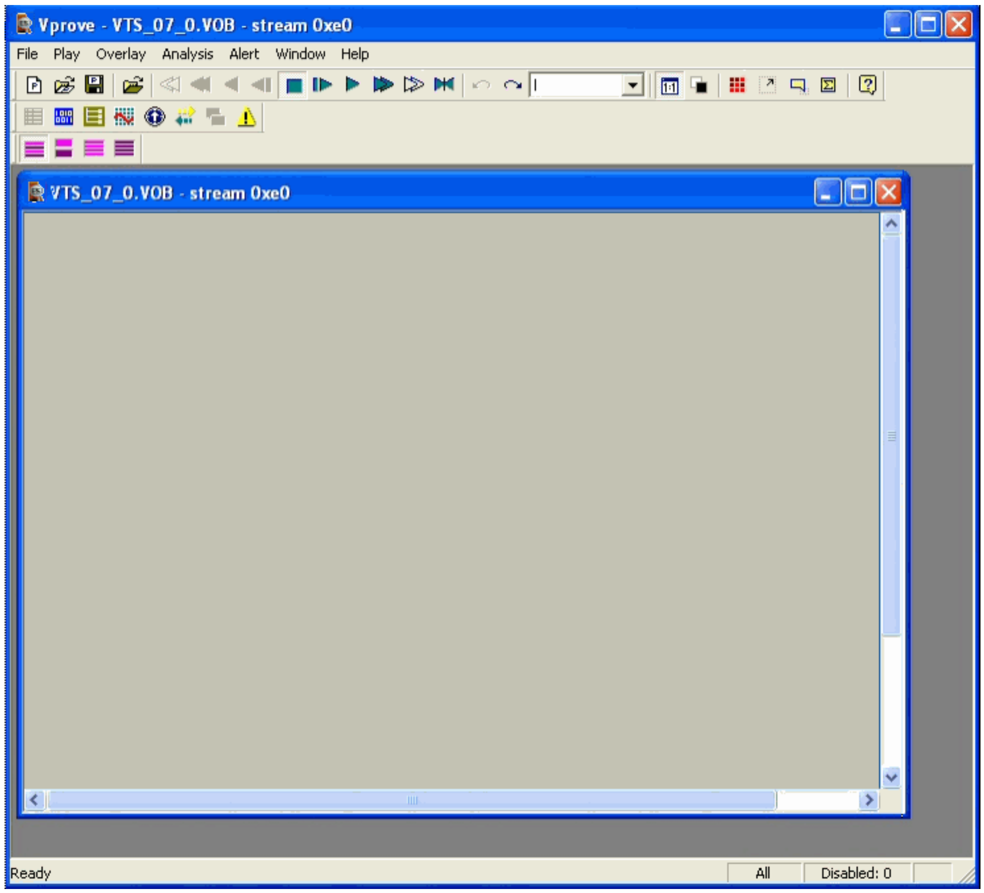
NOTE. *The suggested file name and extension do not have to be accepted -* ***<u>any</u>*** *file name and extension can be entered.*

As well as the video tracks, the other tracks are listed but they cannot be extracted. (This is indicated by the fact that there is no decoder shown in the 'Decoder' column, and the 'Extract' and 'OK' buttons are grayed out if one of these is selected.)

'OK'

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

A video window is then opened with the title of the container file with the track/stream number. In the example below, stream number 0xe0 from an MPEG-2 VOB "VTS_07_0.VOB":



NOTE. *If you decide to play and analyze the selected video stream immediately, then decide to* ***save it later****, select the 'Save as...' option on the 'File' menu. (See File Menu, page 7-13)*

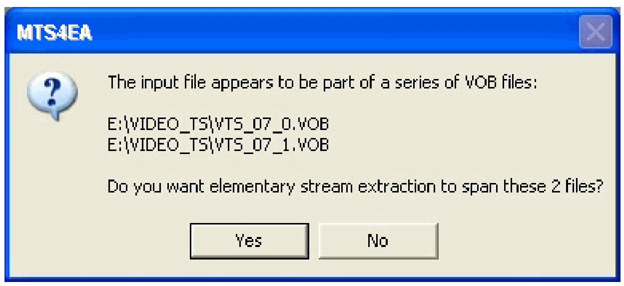
NOTE. *In order to examine the structure of the MP4, 3GPP, MPG (MPEG-2) etc. file, click on the 'View file structure...' option on the 'Analysis' menu (see Analysis Menu, page 7-101).*

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

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

When opening a VOB file, MTS4EA automatically tries to determine if the VOB concerned is one of a series of VOBs, as prescribed by the numbering convention for VOBs, i.e. where the last digit prior to the '.vob' extension is a number from 0 to 9 and indicates how many VOBs there are in a series of VOBs.

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, as described in *Open video…* *Ctrl+O, page 7-13*.

Opening a YUV video file (any file extension). This allows a YUV format file to be viewed.

YUV file format

MTS4EA ignores the extension and does not look for a compressed video standard within the data - it assumes that the data is in the following format:

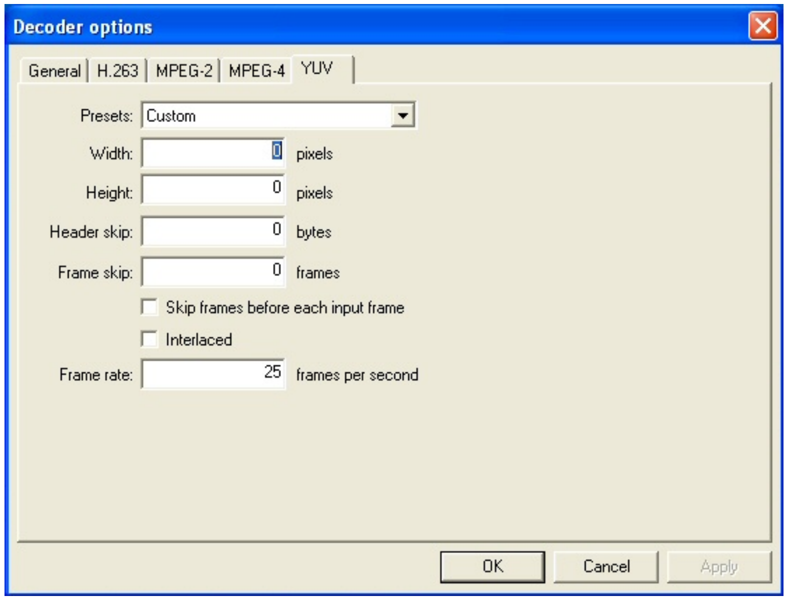
- one byte per sample
- progressive scan (not interlaced)
- row raster order (top picture row first)
- planar YUV 4:2:0 sub-sampled, i.e. 4 bytes of Y data for each byte of U data and each byte of Y data
- Y plane values are 0-255 unsigned
- U and V plane values are unsigned with a DC offset of 128

When opening a YUV format file, by default MTS4EA shows the YUV decoder options, to enter the required parameters.

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

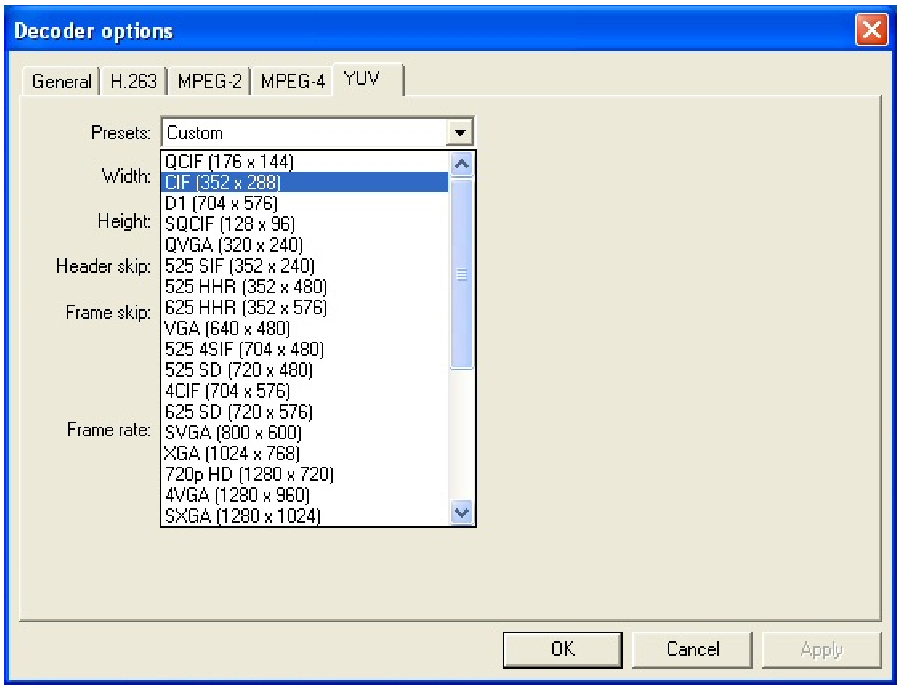
Setting the YUV frame size, etc

Within a YUV file, there is no place to indicate the frame size, frame rate, etc, so when the YUV 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...'):



Automatic reading of frame size/drop-down list (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 above; in any event, the correct values can be entered or selected using the drop-down list:



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

'Interlaced'

Means that the YUV file has interlaced data in the format of complete frames, i.e. 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'

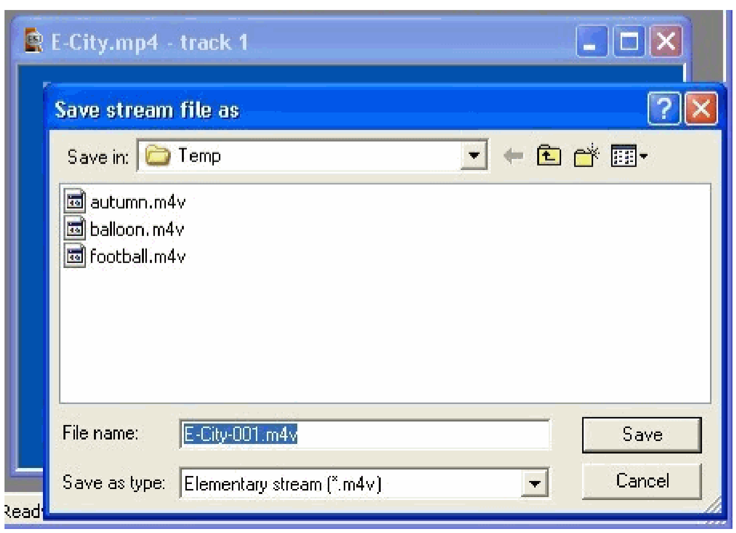
The rate at which to display the YUV frames, in frames per second.

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').*

Save stream as...

This allows the active file in the video window 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) but was not saved at the time it was extracted (See *Open video...* *Ctrl+O*, *page 7-13*):



In this example, the video Elementary Stream from 'E-City.mp4' video track 1 is being saved to a '`.m4v`' file: the '`-001`' (for track 1) and '`.m4v`' are automatically added by MTS4EA (as MTS4EA knows this is an MPEG-4 file: see *Open video...* *Ctrl+O, page 7-13*), although **any** file name and extension can be entered.

New project…

Description of MTS4EA 'projects'. Some of the information saved in a project file includes the:

- video file name being analyzed (and track number, if relevant);
- views open and the screen location of these views;
- location of the tooltips and toolbars
- position the video stream;
- errors which 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 is 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 - i.e. all the settings in '`default.vpp`' are copied into a new project. This has no video file selected.

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

Open project…

(See also New project… , page 7-29.)

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

Save project…

(See also New project… , page 7-29.)

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

Save project as…

(See also New project… , page 7-29.)

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

Close project…

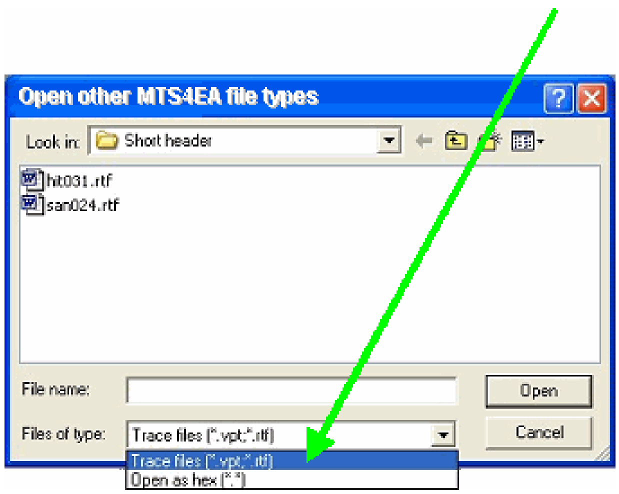
(See also New project… , page 7-29.)

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:

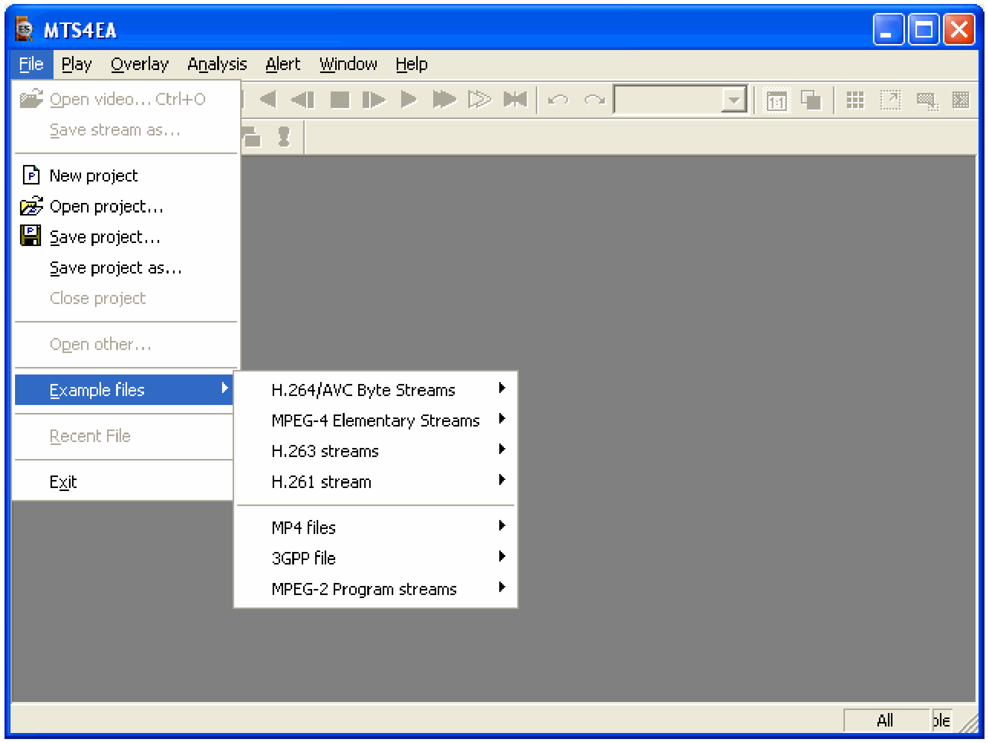
- MTS4EA trace files, with the file extension `.vpt` or `.rtf`
- any other file, to open in the HexView



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 video files which can be played/analyzed in MTS4EA:



These are provided to give examples of various compressed video files to experiment with and to compare with your own compressed video 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 Permitted Video File Types/Formats, page 5-8 and *Standards References*, page 5-14, reference number [13]). Four such files are provided:

Name	H.264 Profile/Level	Warnings/Errors
Neon Night	Baseline/2	- none -
Canary Wharf	Extended/3	use of an invalid (un-initialized) 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)

NOTE. *A YUV reference file is provided for the first 9 frames of the 'Grenadier Guards' bitstream: this YUV reference file can be used for fidelity analysis and visual difference display.*

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

MPEG-4 Elementary Streams. These are MPEG-4 compressed video Elementary Streams (as described in Permitted Video File Types/Formats, page 5-8 and *Standards References*, page 5-14, reference number [1]). 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 i.e. affine model)	`stuffing_bits`, VCV overflow

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

NOTE: *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 Permitted Video File Types/Formats, page 5-8 and *Standards References*, page 5-14, reference number [2]). Three are provided:

Name	Standard/Annexes	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 Permitted Video File Types/Formats, page 5-8 and *Standards References*, page 5-14, reference number [3]).

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 Permitted Video File Types/Formats, page 5-8 and *Standards References*, page 5-14, reference number [11]). Two are provided:

Name	Standard	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

3GPP file. This is a 3GPP `'.3gp'` container file, containing compressed video and other data (as described in Permitted Video File Types/Formats, page 5-8 and *Standards References*, page 5-14, reference number [12]).

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 Permitted Video File Types/Formats, page 5-8 and *Standards References*, page 5-14, reference numbers [14] and [15]).

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 9 frames of the 'Grenadier Guards' bitstream: this YUV reference file can be used for fidelity analysis and visual difference display.*

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

YUV source files for example fidelity analysis. YUV source files used for encoding have been provided for example fidelity analysis with 3 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: note: 25 frames/s)
MPEG-2 (Main Profile/Main Level)	Grenadier Guards	guards_yuv (frames 1-9 only: note: 25 frames/s)
MPEG-4 (Advanced Simple Profile/Level 0)	Man Walking	man_walking_yuv (note: 30 frames/s)

***** the YUV filename is automatically filled in by MTS4EA for these streams - there is no need to enter the filename.

NOTE. *For the 'Grenadier Guards' examples, the YUV of only the first 9 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.yuv [hex]` (this one is opened as YUV video)

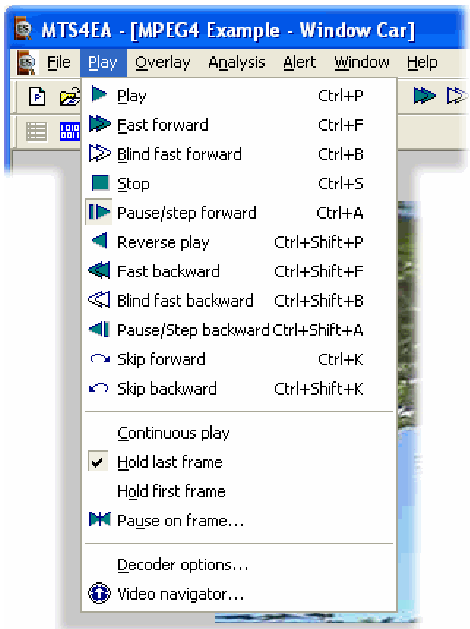
MTS4EA then uses this to determine in which window the file should be reopened.

Exit

This option exits from MTS4EA.

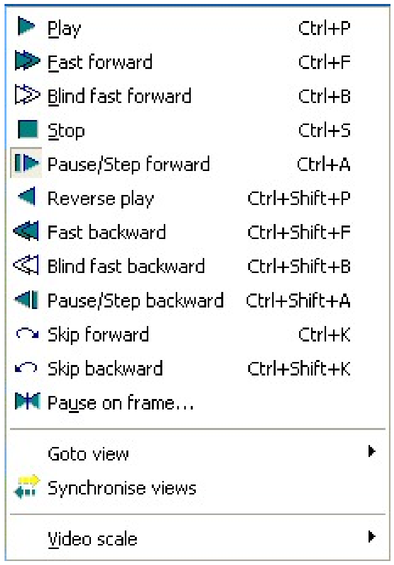
Play Menu

This menu controls the playing of the compressed video:



Right-click pop-up Play menu (video view)

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



The functions of each of these selections are explained below.

Play Ctrl+P

When selected, this plays the compressed video stream until it reaches the end of the data, unless a frame number has been set in 'Pause on frame...' in which case the video will pause at this frame number.

NOTE. *Depending upon the speed of your PC, the video standard 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.)

NOTE. *When playing video forwards or backwards, MTS4EA buffers the video and associated data. By default, MTS4EA allocates a buffer size of 100 MBytes. The amount of buffer required per frame of video and associated data varies considerably with the video size and the video standard. Typically, 100 MBytes 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 Play Menu, page 7-37.)*

At the end of the data the blue background screen is displayed unless:

- 'Hold last frame' is selected, in which case the last frame stays visible, or
- 'Hold first frame' is selected, in which case the first frame is redisplayed.

Fast forward Ctrl+F

When this option is selected the compressed video will be played in ‘Fast forward’ mode, i.e. played more quickly than normal. The speed at which this is replayed depends upon the speed of the computer you are using.

NOTE. *See the <Note> in Play Ctrl+P,* *page 7-38* *regarding a possible buffering delay immediately after this is selected.*

The frame count is given in a box at the top of the video 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 the 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 video is decoded as quickly as possible without displaying it.

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

NOTE. *See the <Note> in Play Ctrl+P, page 7-38 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 video.

The blue background screen is displayed unless:

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

  or

- '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’ .

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

This has two functions:

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

NOTE. *See the <Note> in Play Ctrl+P, page 7-38 regarding a possible buffering delay immediately after this is selected.*

NOTE. *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 video backwards, at normal speed.

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 video backwards, as fast as possible, while still displaying it.

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 video backwards, as fast as possible, without displaying it.

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

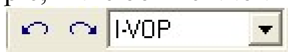
Step backwards through the video, one displayed frame at a time.

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’ - .

See also the notes in Pause/Step forward Ctrl+A, page 7-40.

Skip forward Ctrl+K

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

For example, if the box next to the skip forward icon has 'I-VOP' in it (for example, 

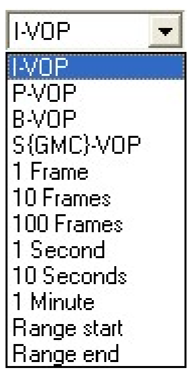
) then clicking the 'Skip forward' icon will skip to the next I-VOP.

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, page 7-38 and Decoder options..., page 7-44 regarding the buffer used to hold the video and associated data. When skipping forward or backward beyond the end of the buffer, MTS4EA must reload 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.



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’ - .

Continuous Play

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

Hold Last Frame

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

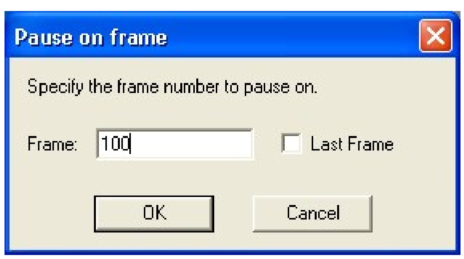
This is selected as **on** by default when MTS4EA starts.

Hold First Frame

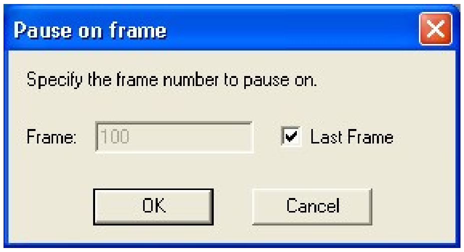
When this option is selected and the video played to the end, the first decoded frame and not the blue background screen, will be displayed.

Pause On Frame 

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 then pause:



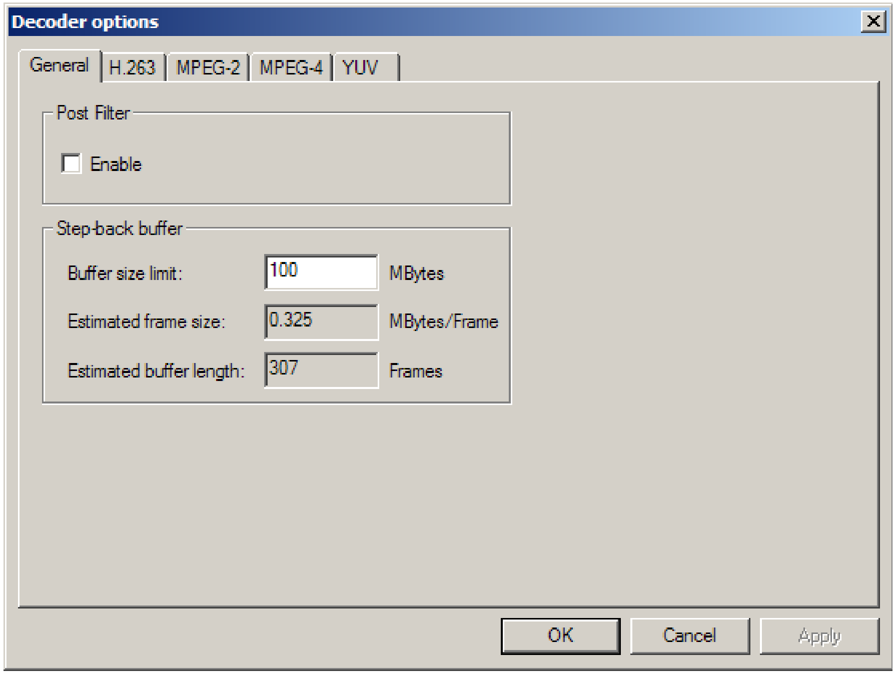
The last frame can be selected by selecting the 'Last Frame' box. When this is done, the video pauses on the last frame. The previously entered frame number (if there is one) is still stored, and may be enabled again by clearing the 'Last Frame' check box:



Decoder options...

Below this selection is a dialog box with various tabs which 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.

General.



<u>Post Filter</u>

This enables proprietary non-linear filtering of the decoded video, which smoothes the block artifacts after decoding. By default, this is switched on.

NOTE. *The Post Filter is disabled by default.*

<u>Step-back buffer</u>

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

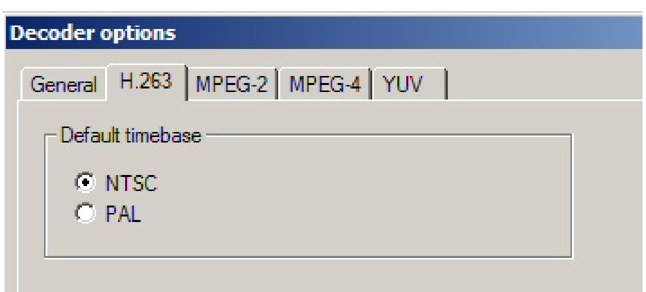
The size of the step-back buffer is shown, with an estimation of how many frames of data can be held within this buffer size.

NOTE. *When playing video forwards or backwards, MTS4EA buffers the video and associated data. By default, MTS4EA allocates a buffer size of 100 MBytes. The amount of buffer required per frame of video and associated data varies considerably with the video size and the video standard. Typically, 100 MBytes 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 Play Menu, page 737.

NOTE. *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 Alerts Menu, page 7-166.*

H.263.



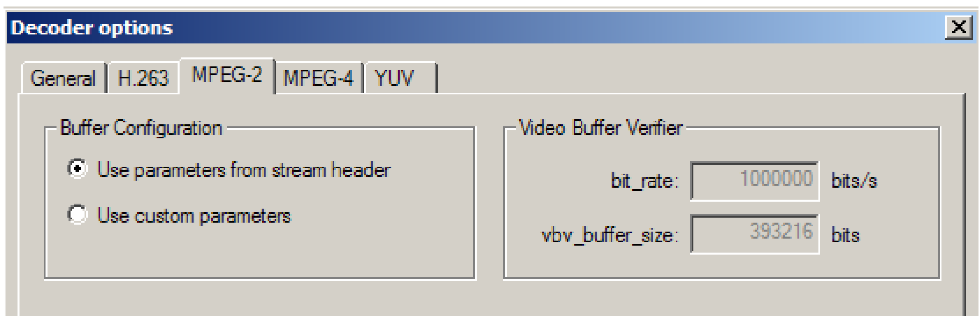
(This has no effect and is grayed out for standards other than H.263.)

NTSC/PAL option

Within the H.263 standard the NTSC standard is assumed, i.e. 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 (that is, at 25 frames per second) does not set these options. In this situation, PAL video would play back at 29.97 frames per second, that is, it would play too quickly. Selecting the 'PAL' option on this tab will play the decoded video at 25 frames per second.

MPEG-2.

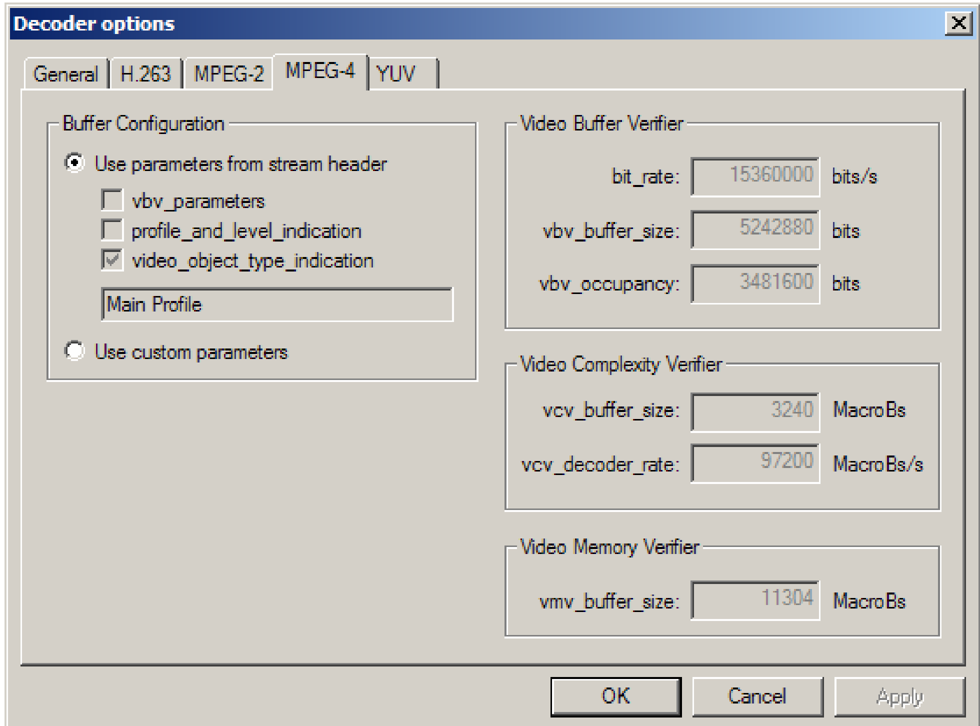


(This has no effect and is grayed out for standards other than MPEG-2)

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

See View buffer analysis..., page 7-151.

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.

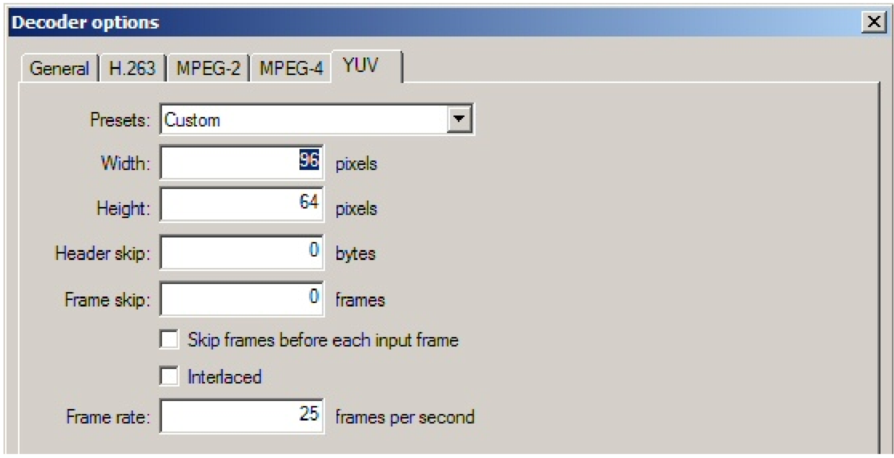
See View buffer analysis..., page 7-151.

YUV. The YUV data is assumed to be in the following format:

- one byte per sample
- progressive scan (not interlaced)
- row raster order (top picture row first)
- planar YUV 4:2:0 sub-sampled i.e. 4 bytes of Y data for each byte of U data and each byte of Y data
- all the Y plane values for a frame are together, followed by the U values for a frame, followed by the V values for a frame
- Y plane values are 0-255 unsigned
- U and V plane values are unsigned with a DC offset of 128.

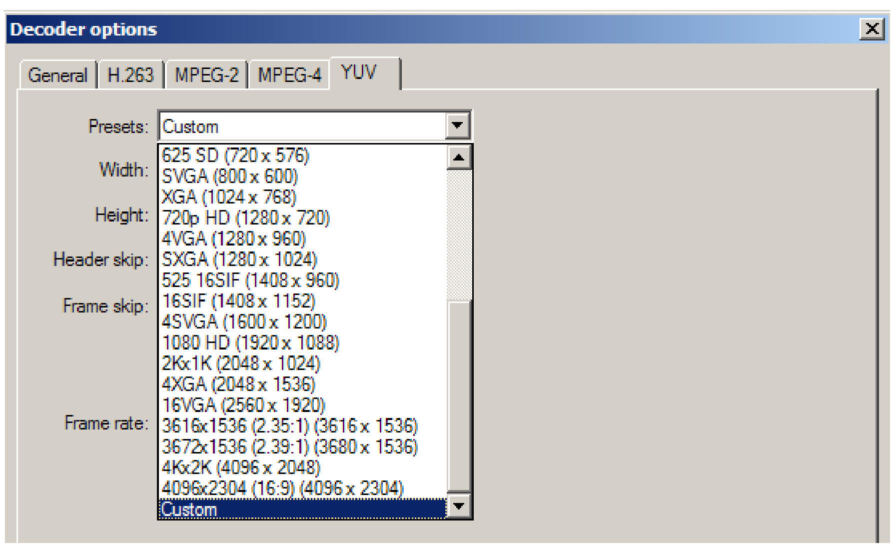
Setting the YUV frame size, etc

Within a YUV file, there is no place to indicate the frame size, frame rate, etc, so when the YUV 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...'):



Automatic reading of frame size/drop-down list (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 above. In any event, the correct values can be entered or selected using the drop-down list:



'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, the frames are skipped after each viewed frame: by selecting the box 'Skip frames before each input frame' then the number of frames are skipped before each viewed frame.

'Interlaced'

Means that the YUV file has interlaced data in the format of complete frames (i.e. 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 YUV frames, in frames per second.

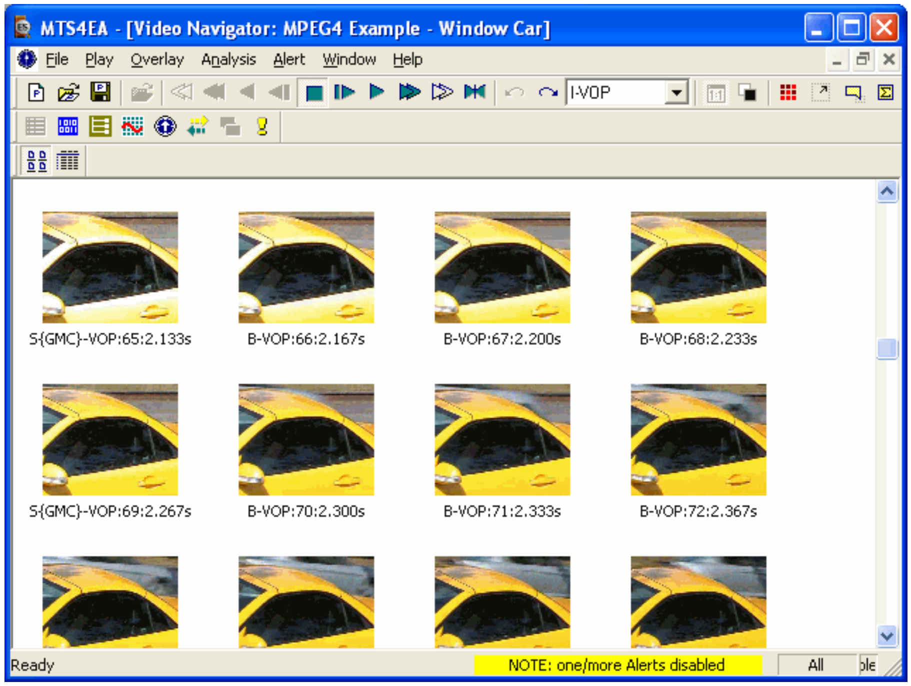
NOTE. *The number entered in the 'Frame rate' field can be an integer (e.g. '30'), or a fraction (e.g. '30000/1001') or a decimal number (e.g. '29.97').*

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).

Thumbnail view.

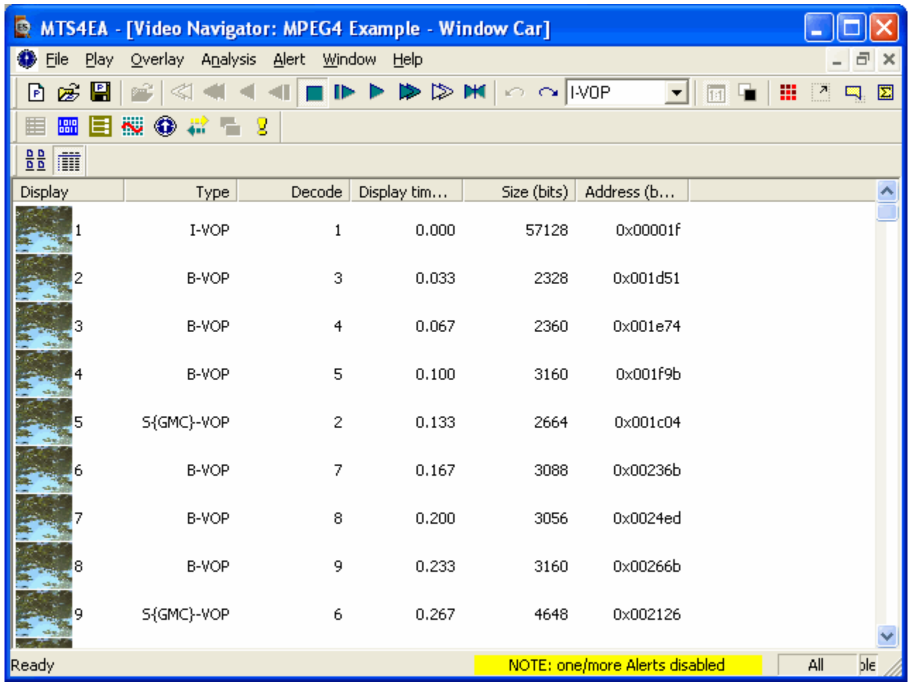


The video frames are shown in display order (decode order can be different if there are B-frames/B-VOPs).

The information given is:

- frame type/VOP type
- display frame/VOP number
- display time in seconds.

Detail view.

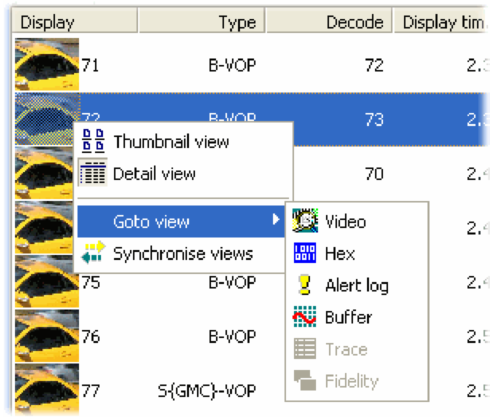


The information given is:

- display frame/VOP number
- frame type/VOP type
- decode frame/VOP number
- display time in seconds
- frame size in bits
- address of start of frame (in the video elementary stream, not in the overall container file if there is one).

The columns may be resized and dragged into a different order if desired.

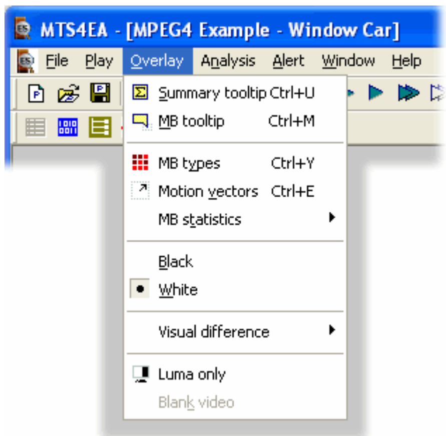
Right-click pop-up menu/Goto views.



See Synchronized views/navigating the views, page 7-6 for more information on navigating between views.

Overlay Menu

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



Summary and Macroblock Tooltips: moving, docking, scroll bars

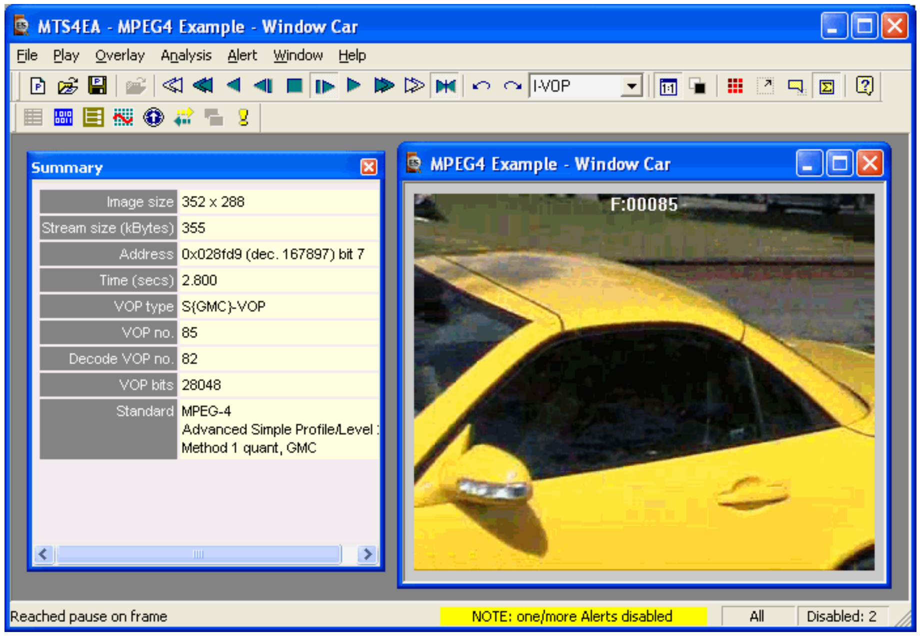
Both the Summary tooltip and the MacroBlock tooltip move and 'dock' in the same way.

HINT. *When the tooltips are dragged near to the edge of the window, they automatically 'dock'.*

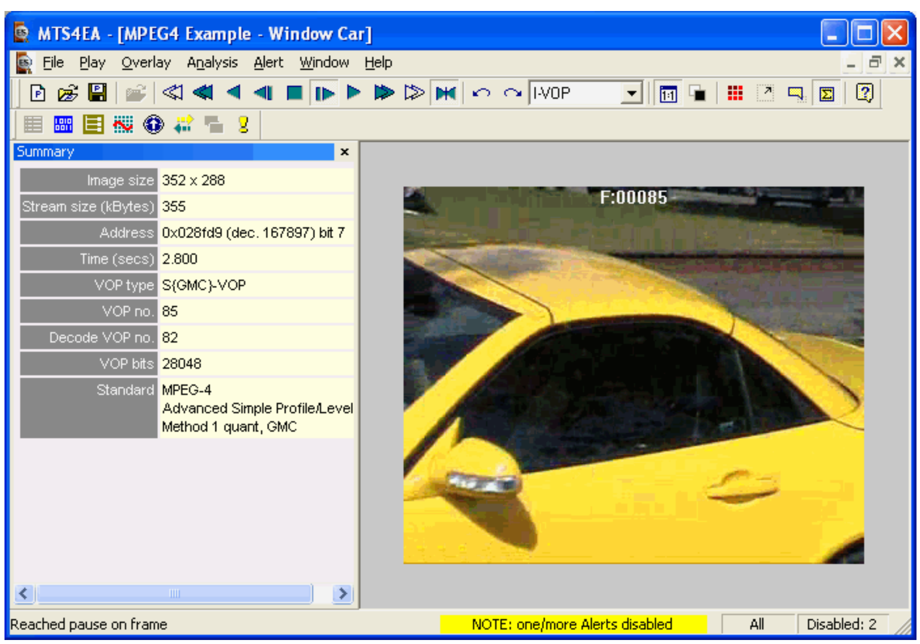
To force undocking, press and hold the <Ctrl> key while dragging with the mouse.

The tooltips bring up independent windows which 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 above):

- 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.

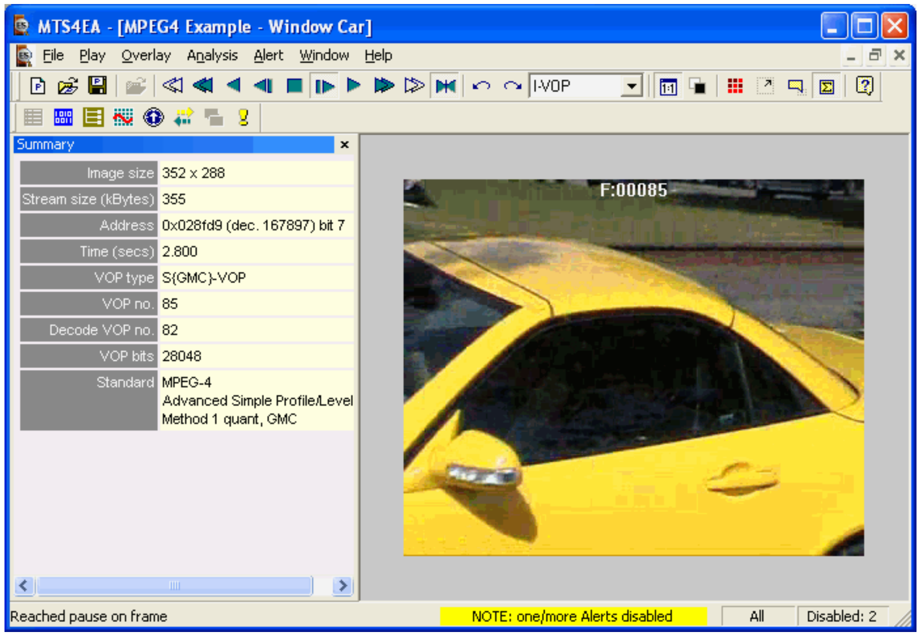
To:

- **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.

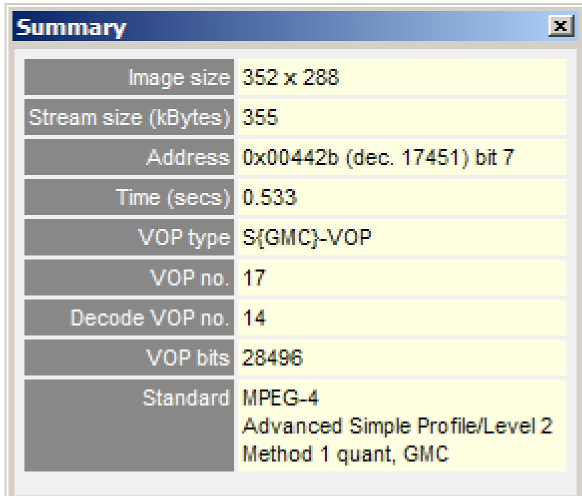
NOTE. *It can sometimes seem hard to undock a tooltip: if necessary, you may need to maximize the MTS4EA window to fill the screen in order for there to be a large enough area for the tooltip to undock into. Try dragging the tooltip outline into the center of the video window. It should then change shape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

HINT. *To force undocking, press and hold the <Ctrl> key while dragging with the mouse.*

Scrolling/scroll bars. When the area available is too small for the whole tooltip to be seen, horizontal or vertical 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 the 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.

Skip and Tref [H.263/H.261 only]. <u>Tref</u>

'**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 (i.e. ~33milliseonds or 40 milliseconds respectively). 'Tref' has a maximum value of 255; when it reaches this value it 'wraps around' back to 0.

<u>Skip</u>

'**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, i.e. ‘Skip’ would equal 2 in this case. (And, for example, if NTSC video were displayed at 10 frames per second then ‘Skip’ would equal 3.)

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 go as follows:

Time (seconds - approx.)	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

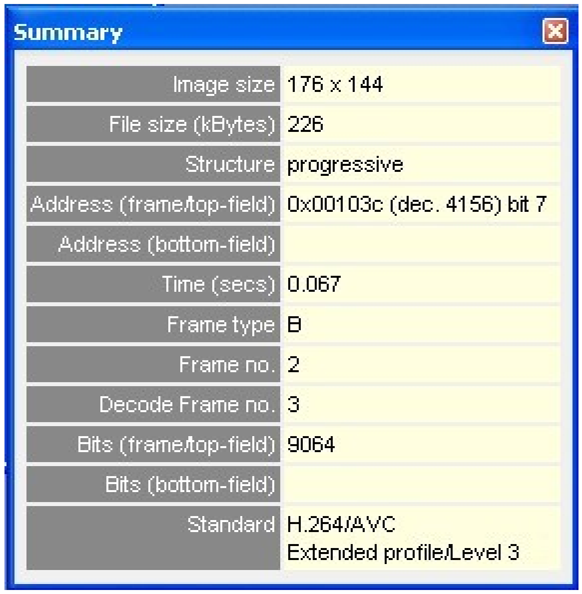
Clearly there are many variations to this; it is often the case that more than one frame is ‘skipped’.

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

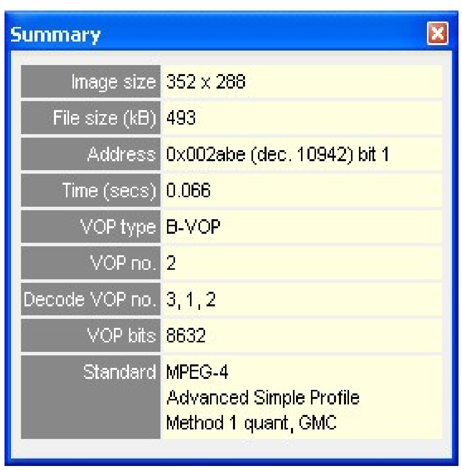
HINT. *Some of the Trace outputs of MTS4EA keep track of the total number of frames skipped* - *this is called ‘Cumul. skip’ (short for Cumulative skip).*

In the last row of the table in the above example Cumul. skip would be 8.

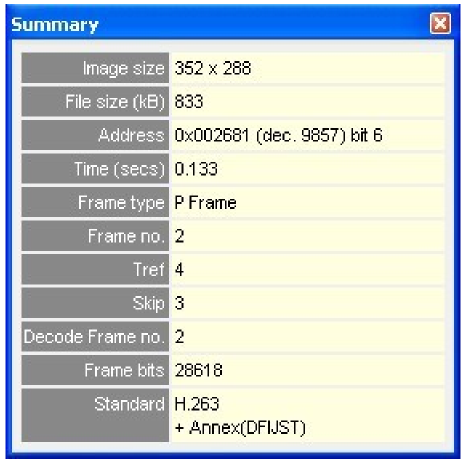
Summary Tooltip varies by video standard. The information provided by the Summary tooltip varies depending upon the video standard that is being analyzed, and the specific options in the video file concerned: the following is an example is for an H.264/AVC stream:



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 Summary Tooltip Ctrl+U, page 7-57.*

<u>Image size</u>

Picture size in pixels, Width x Height (that is, 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)

<u>File size (kBytes)</u>

Size of the video file in KBytes.

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

Structure [H.264/AVC and MPEG-4 Advanced Simple Profile]

For H.264/AVC byte streams and MPEG-4 Advanced Simple Profile, where interlace could be used, this is the structure of the frame being displayed, i.e.:

- progressive, or
- interlaced.

Address or [H.264/AVC and MPEG-4 Advanced Simple Profile] Address (frame/top-field) and Address (bottom-field)

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. Where 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.*

NOTE: *For 'container' files such as MP4 and 3GPP, the address given is the offset from the start of video track which has been selected, NOT the address within the container file (which will be different).*

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 (i.e. occurs first in the stream).

Time (secs)

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

This information is calculated from:

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

- [MPEG-4] the **VOP type**, i.e. I-VOP or P-VOP or B-VOP or S-GMC VOP (see below):
  - **I-VOP:** indicates that this frame is **Intra** coded, i.e. 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 ***or***
  - **P-VOP:** 'P' or 'Predicted' VOP: this frame is **Inter** coded, i.e. partly coded based upon earlier frames ***or***
  - **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 (i.e. not on other B-VOPs) ***or***
  - **S-GMC:** ***(MPEG-4 Advanced Simple Profile only)*** the VOP is an 'S-GMC' VOP which uses Global Motion Compensation (GMC);
- [H.264/AVC] the **Frame type**, i.e. 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, i.e. 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 ***or***
  - **P-frame:** 'P' indicates 'Predicted': this frame is Inter coded, i.e. partly coded based upon earlier frames ***or***
  - **B-frame:** ***(Extended Profile only)*** 'B' or 'Bidirectional Interpolated': this frame is calculated based upon both earlier and later frames ***or***
  - **SI-frame:** 'SI' indicates that this frame is a 'Switching-Intra' coded frame, i.e. there is a switch between two different streams at this point ***or***
  - **SP-frame:** 'SP' indicates that this frame is a 'Switching-Inter' coded frame, i.e. 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.e. I-frame or P-frame or B-frame (see below):
    - **I-frame:** ‘I’ indicates that this frame is Intra coded, i.e. 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 ***or***
    - **P-frame:** ‘P’ indicates ‘Predicted’: this frame is Inter coded, i.e. partly coded based upon earlier frames ***or***
    - **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.e. I-frame or P-frame or B-frame (see below):
    - **I-frame:** ‘I’ indicates that this frame is Intra coded, i.e. 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 ***or***
    - **P-frame:** ‘P’ indicates ‘Predicted’: this frame is Inter coded, i.e. partly coded based upon earlier frames.

VOP no. [MPEG-4]/Frame no. [H.264/AVC, 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.

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 Summary Tooltip Ctrl+U, page 7-57.

Decode VOP no. [MPEG-4]/Decode Frame no. [H.264/AVC, MPEG-2]

The data which appears in these fields differ from the video standard.

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

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

That is, for sequences with B-VOPs the order in which the VOPs/frames are stored in the video file is different 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 to the decode order in every VOP after the first B-VOP occurs in the bitstream (in the display order).*

<u>VOP bits [MPEG-4]/Bits (frame/top-field) and Bits (bottom-field) [H.264/AVC]/Frame bits [MPEG-2, H.263, H.261]</u>

The number of bits used in that:

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

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

- [MPEG-2, H.263, H.261] frame

<u>[Fidelity analysis] PSNR 255 / PSNR BT.601 / RMSE / MSE / MAD / SAD</u>

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.

Standard

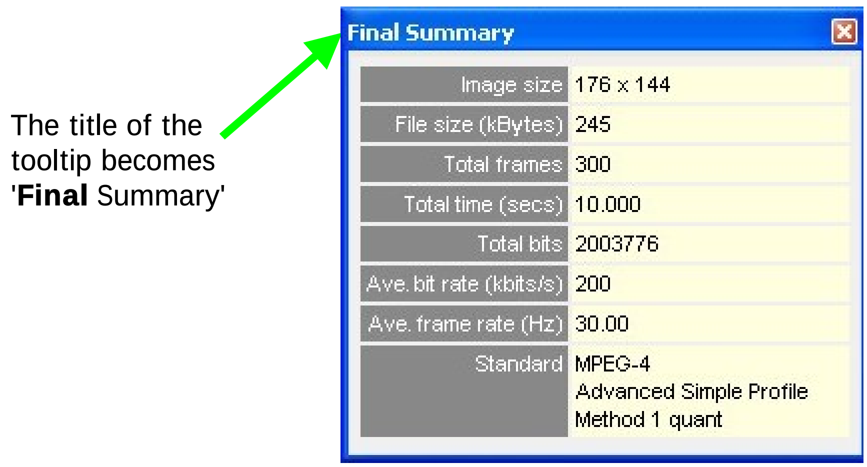
The video standard used, with various additional information as appropriate, such as:

- [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]
    - 'Profile' and 'Level';
- [H.264/AVC]
    - 'Profile' and 'Level'
    - entropy coding mode, i.e. ‘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 title bar of the summary tooltip displays 'Final Summary':



The meanings of:

- Image size
- File size (kBytes)
- Standard

are the same as given in *Summary Tooltip*  *Ctrl+U, page 7-57*.

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; that is, the MTS4EA keeps track of the playing time, irrespective 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 which 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)

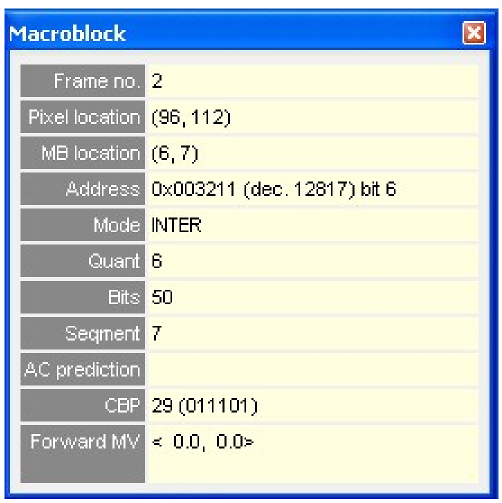
This is 'Total frames' divided by 'Total time', in frames/second (Hz).

[Fidelity analysis] Ave. PSNR 255 / PSNR BT.601 / RMSE / MSE / MAD / SAD

When fidelity analysis is enabled, the average of the selected fidelity analysis values across all the VOPs/frames is displayed.

MB Tooltip Ctrl+M

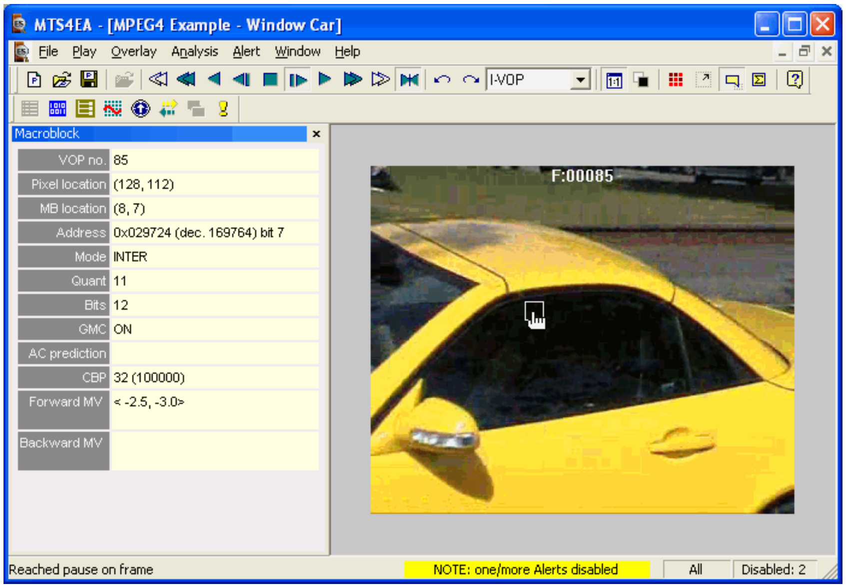
(MacroBlock Tooltip) This displays information about a selected MacroBlock:



NOTE: *The information displayed by the MB tooltip varies with the video standard and the options used within the particular video sequence - see MB Tooltip Ctrl+M, page 7-67.*

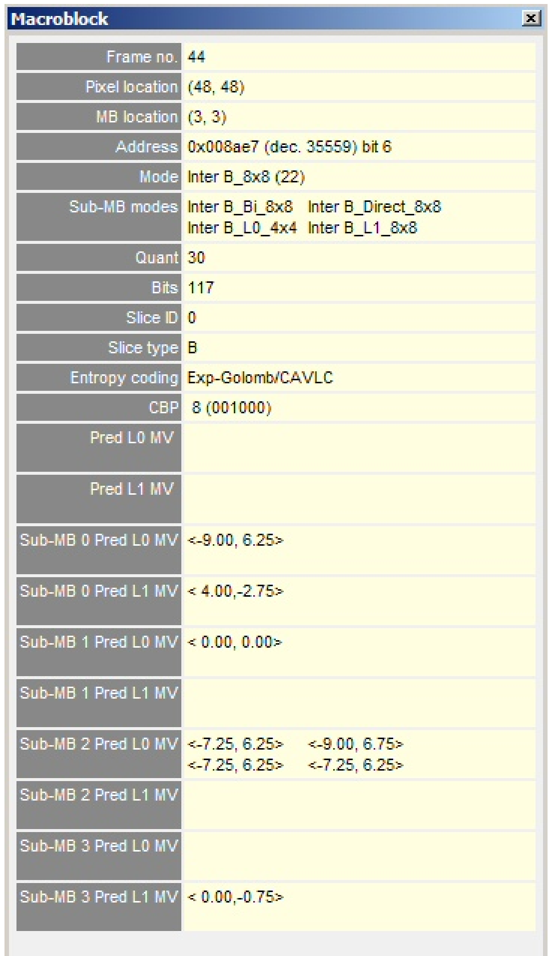
The example above is for an MPEG-4 sequence, and is examining an 'Inter' MacroBlock.

The MacroBlock selected is signified by a white delineation box around the 16x16 MacroBlock border. The MacroBlock is selected simply by hovering the cursor (hand) over a block.

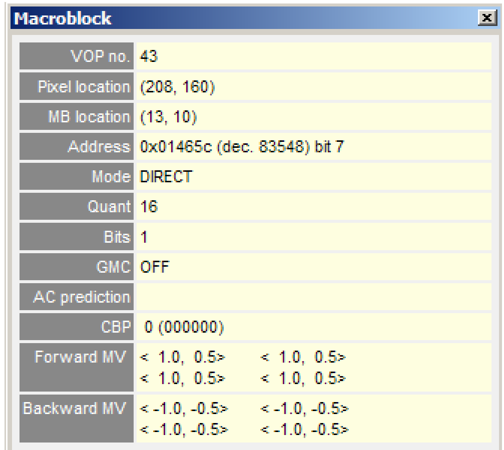


Viewing the MacroBlock data in other views. To see the MacroBlock data in other views - e.g. the HexView - right click at the MacroBlock location of interest and select 'Goto view' from the context menu, then select one of the views. For example, if Hex View is selected, the first byte of the selected MacroBlock is displayed in the HexView.
See Starting to Use MTS4EA, page 7-3.

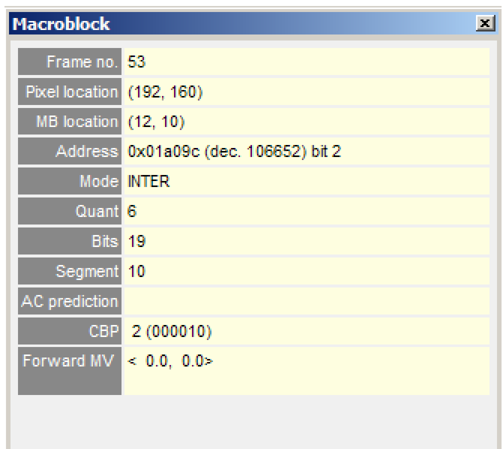
MB Tooltip varies by video standard. The information provided by the MB tooltip varies depending upon the video standard that is being analyzed, and the options used in the video file:



H.264/AVC Inter MB



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



H.263 Inter MB (note 'Segment')

Information provided by MB Tooltip.

<u>VOP no. [MPEG-4]/Frame no. [H.264/AVC, MPEG-2, H.263, H.261]</u>

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 *Summary Tooltip Ctrl+U, page 7-57* for more information on decoded versus displayed VOP/frame numbers.

Pixel location

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

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

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.*

NOTE: *For 'container' files such as MP4, 3GPP and MPG (MPEG-2), the address given is the offset from the start of video track which 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 (that is, occurs first in the stream).

Frame/field coding [H.264/AVC, 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 3 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 2nd and 3rd parts of the Data Partitioned MacroBlock data respectively.

<u>Mode</u>

The MacroBlock type/coding mode as given in MB Types Ctrl+Y, page 7-77.

<u>Sub-MB modes [H.264/AVC only]</u>

The type/coding mode of the blocks within the MacroBlock as given in MB Types Ctrl+Y, page 7-77.

<u>Quant</u>

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).

<u>Bits</u>

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 3 parts of the Data partitioned data for the MacroBlock.

<u>GMC [optional, MPEG-4 Advanced Simple Profile only]</u>

This denotes if the MacroBlock uses Global Motion Compensation (GMC) in the calculation of motion vectors. This is either:

- **ON -** GMC is on, or
- **OFF -** GMC is off.

<u>Segment [H.263, H.261 only]</u>

The segment number, representing the GOB number or slice (starting at zero from the top of the frame). See MB statistics..., page 7-83.

<u>Slice [MPEG-2 only]</u>

The number of the slice (starting at zero from the top of the frame).

<u>Slice ID [H.264/AVC only]</u>

The number of the slice (starting at zero from the top of the frame).

<u>Slice type [H.264/AVC only]</u>

The slice type, i.e. 'I' or 'P' or [Extended Profile only] 'B' or 'SI' or 'SP'.

Entropy coding [H.264/AVC only]

The entropy coding mode of the slice, i.e. CAVLC.

Transform [MPEG-4 Advanced Simple Profile only]

Shows the DCT type: ‘field’ or ‘frame’, as detemined by the value of dct_type in MPEG-4 ASP interlaced_information().

AC prediction [MPEG-4, H.263 and H.261 only]

This is used differently for MPEG-4 and for H.263:

- MPEG-4:
  - **ON -** AC prediction is on, or
  - **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 and AC prediction
  - **2 -** Horizontal DC and AC prediction.

CBP

Coded Block Pattern: this is a number from 0-63, signifying which of the six 8x8 blocks that comprize 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]

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, that is, they have been predicted forward from a preceding VOP/frame.)

[MPEG-4 Advanced Simple Profile] For interlaced MacroBlocks, this shows which fields the forward Motion Vectors predict from, as determined by the value of forward_top_field_reference and forward_bottom_field_reference in MPEG-4 Adv. Simple Profile interlaced_information(). This is displayed as follows:

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) and MPEG-2 only]

These only appear for B-VOPs in MPEG-4 Advanced Simple Profile and MPEG-2 B-frames: 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, i.e. 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 only]

The Motion Vectors for the MacroBlock. Data only appears in this field if there is no sub-division of the MacroBlock:

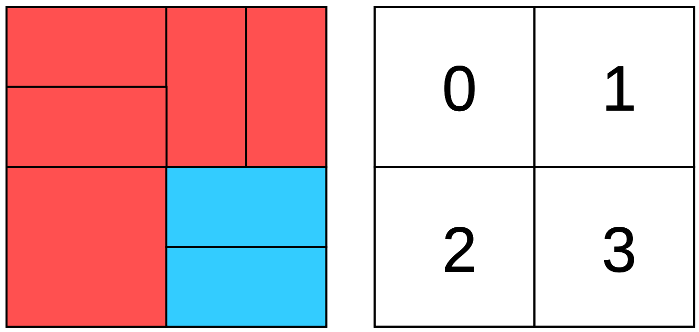
- 'L0' refers to List 0 reference frames (that is 'forward' reference, i.e. from past frames: as an example 'P' type MacroBlocks can only be List 0);
- 'L1' refers to List 1 reference frames (that is 'forward' and 'backward' reference, i.e. from past and future frames: as an example 'B' type MacroBlocks can be List 1 or List 0).

Sub-MB 0/1/2/3 Pred L0/L1 MV [H.264/AVC only]

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).

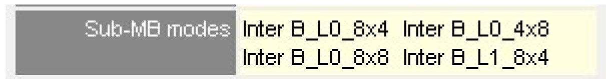
The data provided here follows the standard and is shown in the layout as given in the 'Sub-MB modes' field of the MacroBlock tooltip and MB types overlay.

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.

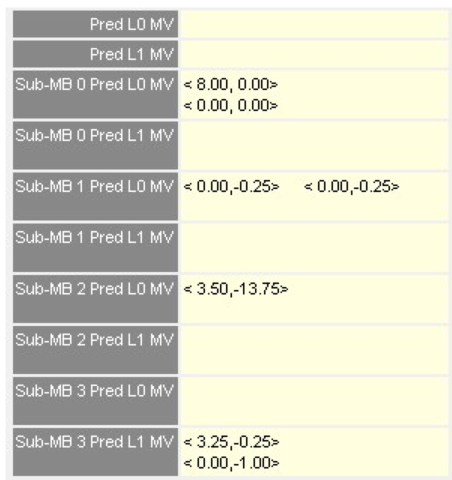
Then the 'Sub-MB modes' field would look like this:



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:



where:

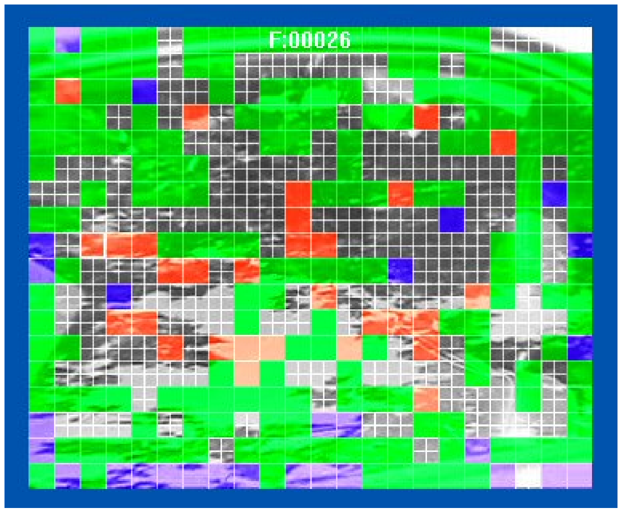
- Pred L0 MV and Pred L1 MV both have no data, as the MacroBlock is sub-divided;
- Sub-MB 0 Pred 0 MV has the data, as this is a List 0 block, organized into two pairs of MVs, one above the other, as the block is divided into two 8x4 sub-blocks, one above the other;
- Sub-MB 1 Pred 0 MV has the data, as this is a List 0 block, organized into two pairs of MVs, side by side, as the block is divided into two 4x8 sub-blocks, larranged side by side;
- Sub-MB 2 Pred 0 MV has the data, as this is a List 0 block, organized into one pair of MVs only, as the block is a single 8x8 block;
- Sub-MB 3 Pred 1 MV has the data, as this is a List 1 block, organized into two pairs of MVs, one above the other, as the block is divided into two 8x4 sub-blocks, one above the other.

[Fidelity analysis] PSNR 255 / PSNR BT.601 / RMSE / MSE / MAD / SAD

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

MB Types Ctrl+Y

(MacroBlock Types)



(The above example is a B-VOP of an MPEG-4 Advanced Simple Profile stream)

This option 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, page 7-205 for more information).

The lines denote the edges of the MacroBlock, or where the MacroBlock has 4 motion vectors, then the MacroBlock is also divided into 4 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]), shown in the following colors:

MB Type	Prediction mode	Table Index	Sub-MB Types	Color	Slice type(s)
I	Intra 4x4 and 16x16	0 - 24	-	Green	I, P, B, SI, SP
I	Intra inferred	25	-	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

MPEG-4. MPEG-4 has the following possible coded MacroBlock types (see reference [1], Table B-1), 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]), 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

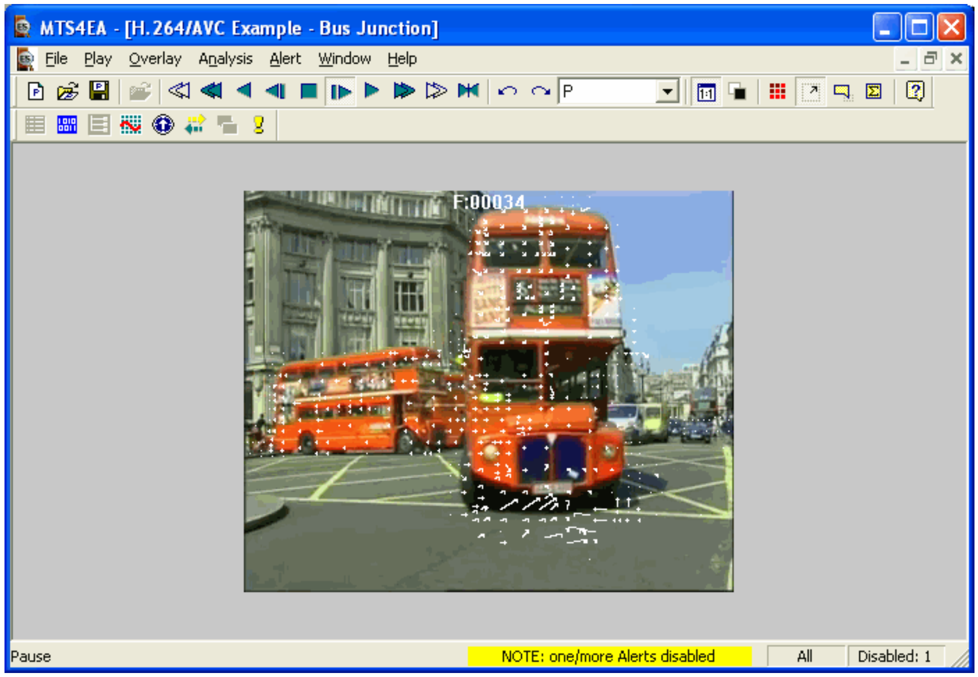
H.263+ and H.263. H263 has 6 possible coded MacroBlock types (see reference [2], Table 9/H.263), shown in the following colors:

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/H261), 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

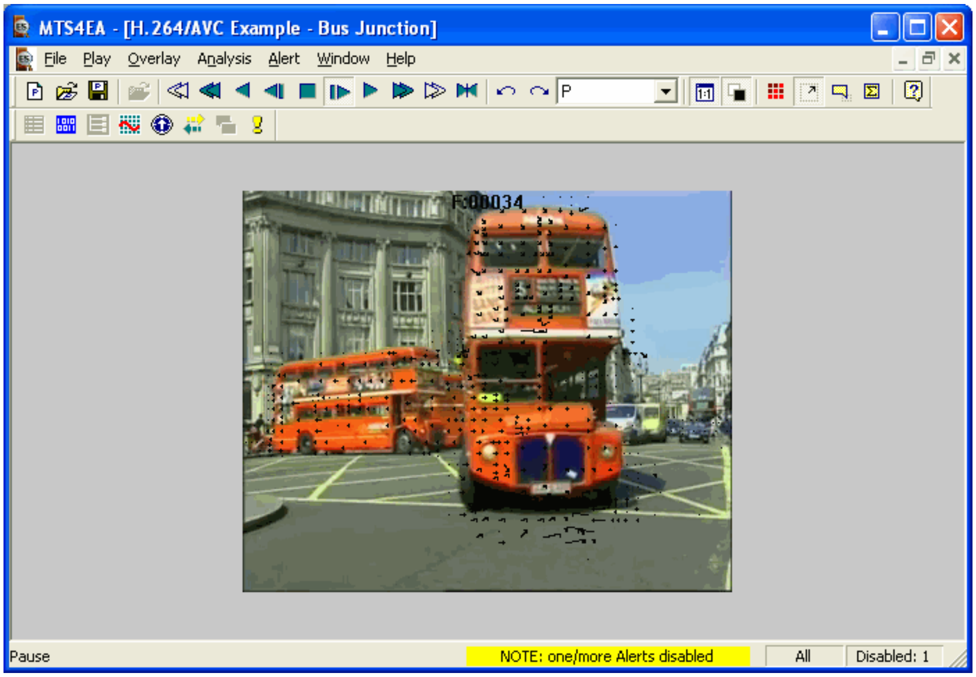


The motion vectors used in relevant video frames can be displayed on the frame currently being viewed. They are switched on and off either:

- from the 'Overlay' menu, 'Motion vectors' option, or
- by clicking the toolbar icon , or
- by pressing 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 , page 7-98) for all frames except B-VOPs/B-frames - see Motion Vectors Ctrl+E, page 7-80 for information on this.

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

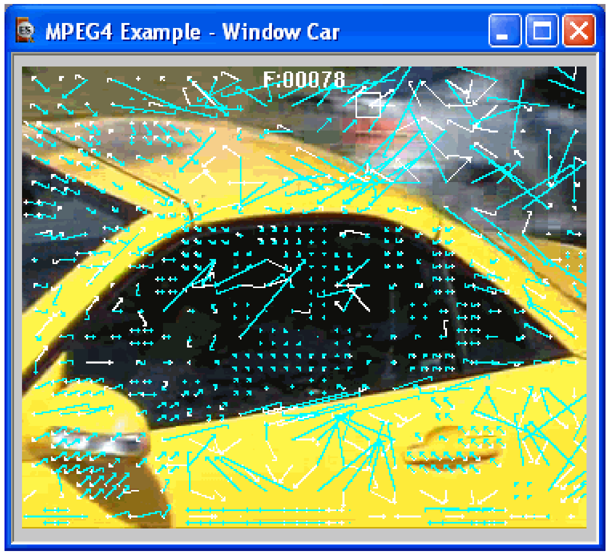


The lengths of the motion vectors are plotted proportionally to the actual motion.

The vector has an arrow-head 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 which 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. Forward motion vectors are displayed in white (or black); Backward motion vectors are displayed in light blue (or dark blue):



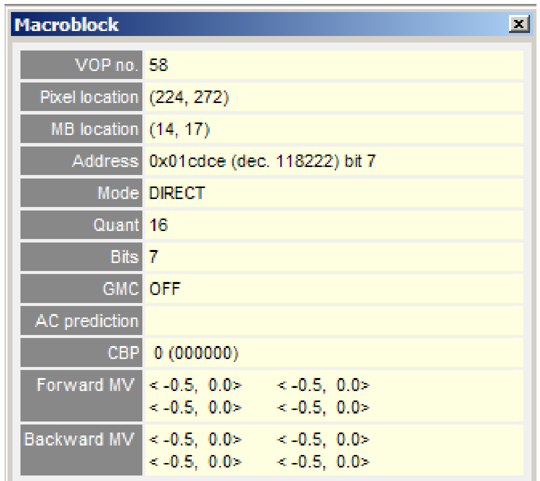
In **Interlaced** video sequences the motion vectors are displayed in the following colors:

MV type	Field	MV color - out	MV color - in
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 MB Tooltip Ctrl+M, page 7-67.

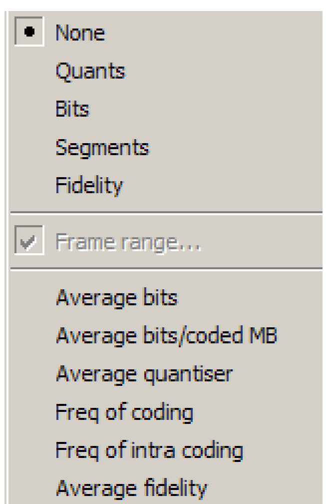
For MacroBlocks with 4 (or more) motion vectors, the MacroBlock tooltip is expanded with extra information to display all the motion vectors: the example below is for a MacroBlock in a B-VOP (in MPEG-4 Advanced Simple Profile) where there are 4 Forward and 4 Backward motion vectors:



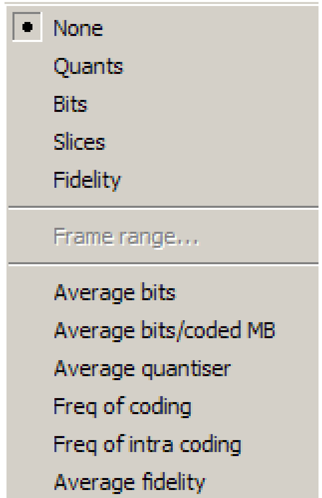
MB statistics...

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

The menu varies (slightly) with the video standard:



MPEG-4, H.263, H.261



H.264/AVC, MPEG-2

NOTE. *To view the total sequence statistics in relation to the picture content, use this option with the 'Hold last frame' option in the Play menu.*

HINT. *For interlaced video streams, some of the MB statistics overlays are not visible if the video is displayed in 'combined view' (i.e. frame view). To see the statistics in this situation, switch to the separated fields view using the interlace toolbar (see Interlace toolbar, page 7-201).*

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: *'Frame range...' is unavailable (grayed out) in pause mode (indicated when the Pause/Step forward icon or Pause/Step backward icon is pushed in on the toolbar and 'Play' menu). This is because the frame range over which statistics are to be accumulated cannot be changed during pause mode.*

This is because the MTS4EA could not otherwise verify 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.

NOTE: *When a range of frames has been set, the 'Frame range...' item on the MB statistics menu changes, with a tick to indicate a range of frames has been set (see MB statistics..., page 7-83).*

NOTE: *The status bar indicates when the video frame being displayed is inside or out of the selected range of frames (see MB statistics..., page 7-83).*

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

NOTE: *It 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 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, even though in MPEG-4 and H.263 a single bit is sent to signify a 'not coded' MacroBlock.

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

NOTE. *The Bits option can be used with Motion Vectors and/or MacroBlock types, but not with any other digit overlay option.*

Segments [MPEG-4, H.263, H.261].

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

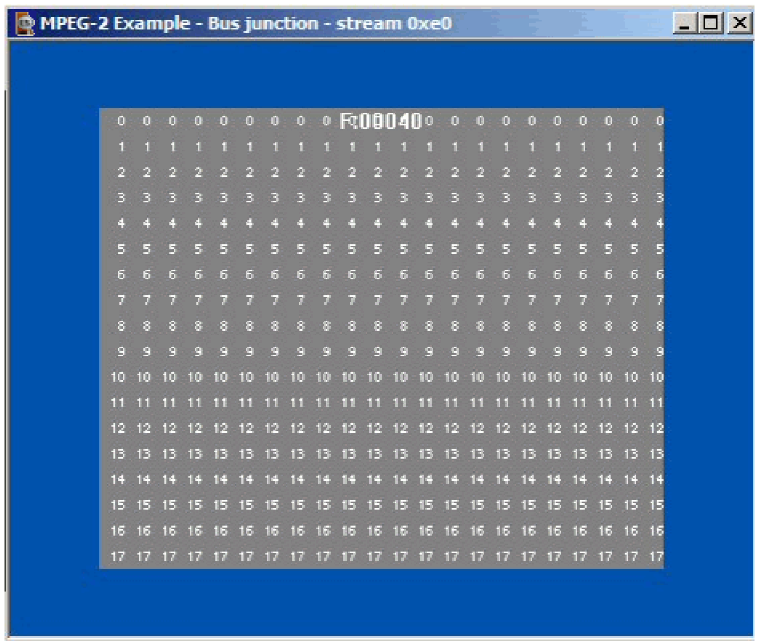
This option shows the segment numbering in use for the current frame. Segments are defined as being Group of Blocks (if GOB headers are sent in H.263) or Slices (as defined in Annex K of H.263+). If a segment number is zero then it is not displayed - hence if no GOB headers or Slices are sent, then this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



(In order to see the slice ID numbers more clearly the video has been blanked, using the menu selection 'Overlay - Blank video' .)

The slice ID overlay option can be used with Motion Vectors and/or MacroBlock types, but not with any other digit overlay option.

Fidelity.

NOTE. *'Fidelity' is grayed out until fidelity analysis is enabled; see Fidelity enable..., page 7-132.*

This option shows the fidelity analysis values for each MacroBlock, in dB (decibels):

The fidelity analysis done (i.e. which type of PSNR, etc analysis) is given in the 'Fidelity' tab of the Analysis options - see Fidelity enable..., page 7-132 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.

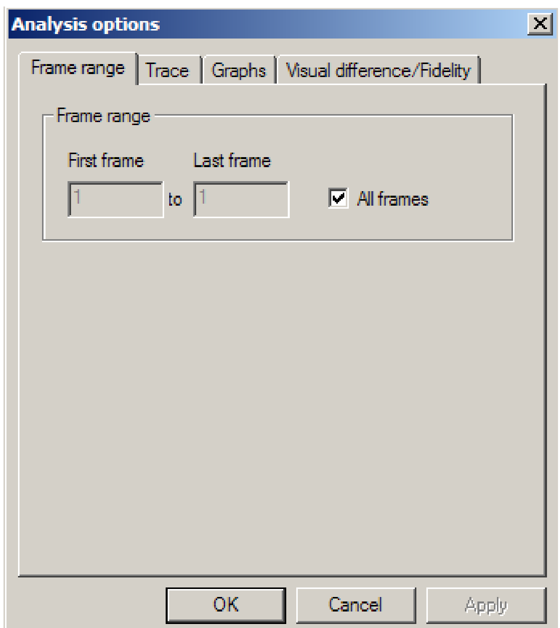
NOTE. *The frame range set on this tab is the same frame range as used for the Trace and Graph analysis (see Trace enable Ctrl+T, page 7-102).*

NOTE. *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. *'Frame range...' is unavailable (grayed out) in pause mode. (indicated when the Pause/Step forward icon or Pause/Step backward icon is pushed in on the toolbar and 'Play' menu). This is because the frame range over which statistics are to be accumulated cannot be changed during pause mode.*

This is because the MTS4EA could not otherwise verify 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:

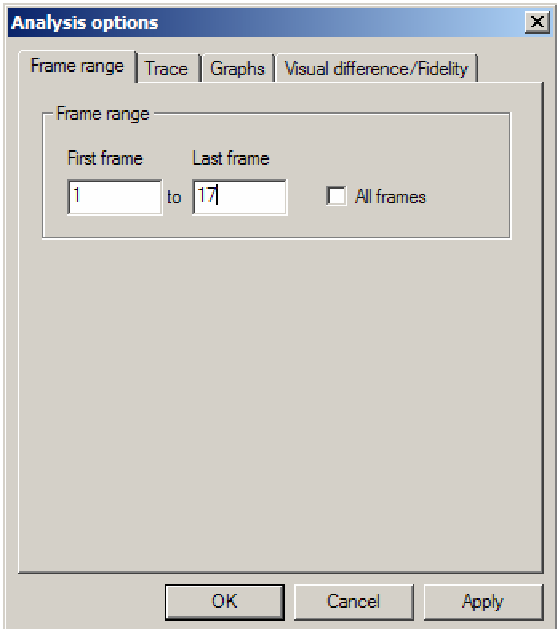


All frames

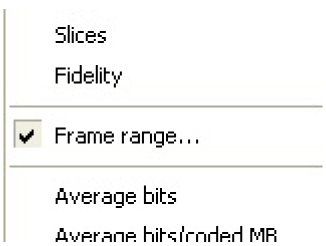
This option is the default and calculates the sequence statistics from the start of the sequence until the current frame.

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):



NOTE. *When a range of frames has been set, the 'Frame range...' item on the MB statistics menu changes, with a tick 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:

When there is no range set (i.e. all frames)

The frame counter text is in black with a gray background:

When a frame range is set

When a frame range has been set and the current displayed frame is within the range specified, the frame counter background changes to green, as below:

When a frame range has been set but the current displayed frame is outside of the range set, the frame counter changes to white text on a red background:

Average Bits. This option displays the average number of bits used to code each MacroBlock, by dividing the total accumulated bits for each MacroBlock by the total number of frames in the frame range.

Both coded and not coded MacroBlocks are included in the average bits statistics.

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

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, i.e. 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.

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

Average Quantizer. This option displays the average quantizer used to code each MacroBlock over the frame range specified. It is displayed to one decimal place.

Freq of Coding. This option displays the 'Frequency of Coding' i.e. number of times each MacroBlock has been coded over the frame range specified.

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

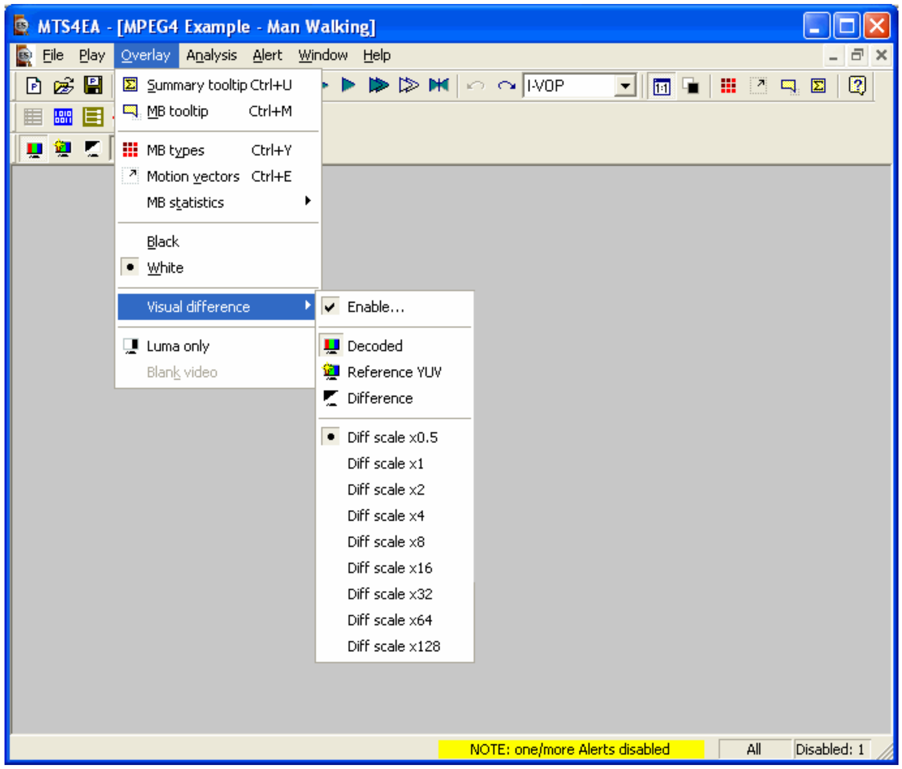
Freq of Intra Coding. This option displays the number of times each MacroBlock has been coded in Intra Mode over the frame range specified.

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

Average fidelity. This option shows the average fidelity analysis values for each MacroBlock, in dB (decibels), over the selected frame range.

The fidelity analysis done (i.e. which type of PSNR, etc analysis) is given in the 'Fidelity' tab of the Analysis options - see Fidelity enable..., page 7-132 for more information.

Visual difference



This menu selection enables the 'visual difference' video display; this uses a YUV reference file to show a video view of:

- the encoded (compressed) bitstream, or
- the YUV reference bitstream, or
- the visual difference between the encoded bitstream and the YUV 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 9 frames of the YUV reference file are provided for the H.264/AVC and MPEG-2 bitstreams 'Grenadier Guards'.*

HINT. *The visual difference view can be used to display the difference between two YUV files: to do this, open the first YUV file using the standard 'File-open' (selecting 'Open as YUV') and then select the second YUV as the reference file.*

When displaying the compressed bitstream or the YUV 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 YUV files and therefore will be grayed out).

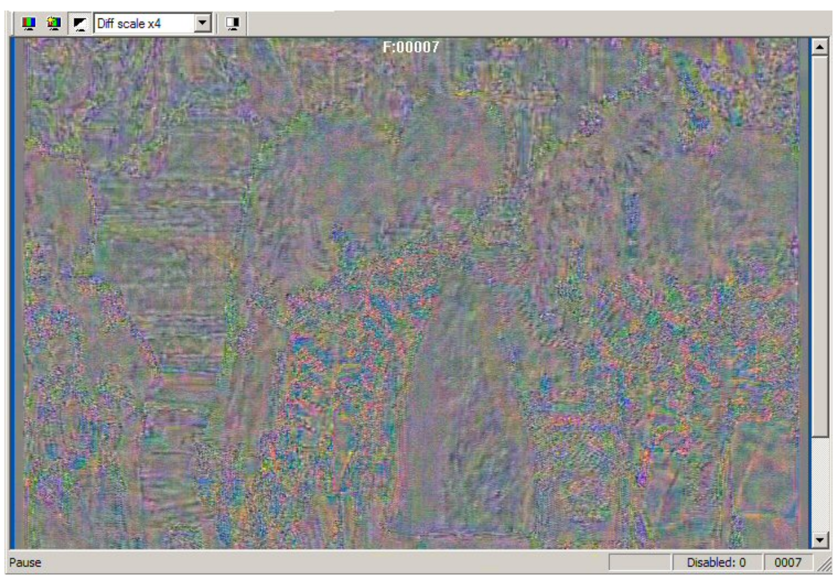
Visual difference icon toolbar. This toolbar is displayed when the visual difference view is enabled:

The functions of the icons are shown below in *Visual difference*, page 7-93.

Show encoded (compressed). Selecting this menu option (or clicking on the  icon) displays the standard video window, i.e. shows the compressed bitstream which has been decoded by MTS4EA:

Show YUV reference. Selecting this menu option (or clicking on the  icon) displays the frame in the YUV reference file which corresponds most closely in time to the corresponding frame in the compressed bitstream:

Show difference.



This subtracts the YUV reference video from the compressed video, frame-by-frame.

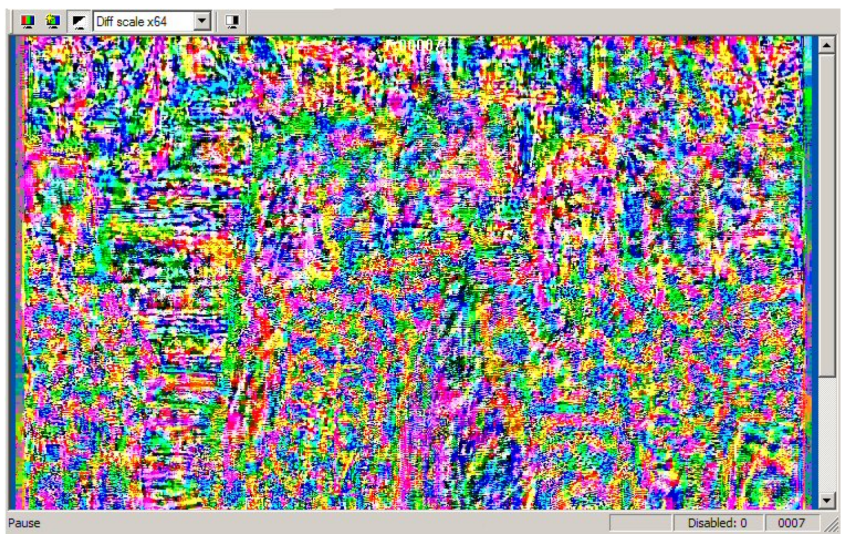
MTS4EA uses the timing given in the:

- compressed bitstream itself for the displayed frame times of the compressed bitstream;
- YUV reference setup tab of MTS4EA (i.e. the frame rate) for the YUV reference file.

When doing the subtraction, MTS4EA uses the YUV reference file and the corresponding frames from the compressed bitstream which are closest to each other in time.

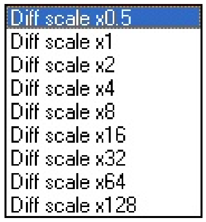
NOTE. *Therefore if the frame rate of the YUV reference file is set incorrectly, the visual differences will invariably appear to be much greater than they should be.*

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:



Show luma only.

This allows the luma (luminance) only of the compressed bitstream, or YUV reference, or difference to be displayed.

Black/White 

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 MB statistics..., page 7-83 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.

Show luma only

When this option is selected, only the luminance is shown of the image in the video window:

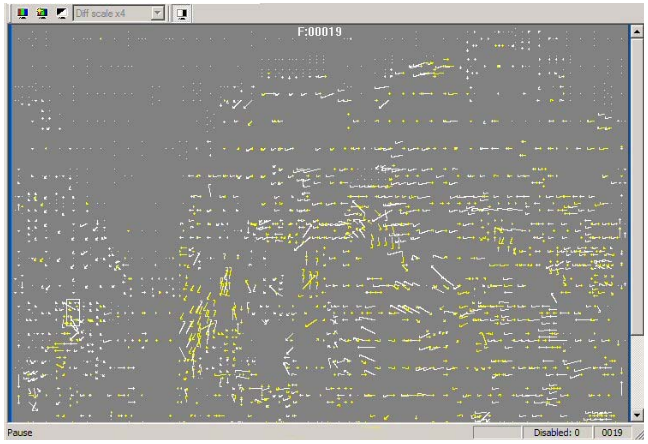
luma only not selected

luma only selected

Blank Video

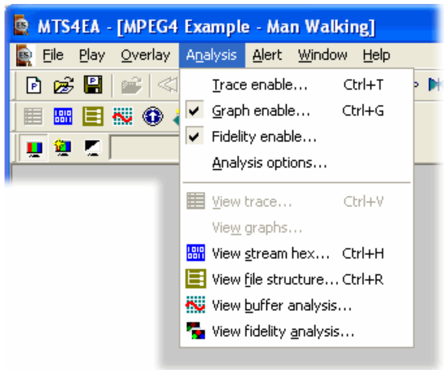
This option will blank the video output to a gray color. It is included so that the motion vectors and other data can be viewed more clearly without the visual interference of the decoded image.

In the example below, the motion vector overlay is switched on and the video is blanked:



NOTE. *The 'Blank video' option is only available when one of the overlays on the Overlay menu is being used, otherwise it is grayed out (because there would be no need to blank the video if there were no overlay data to see).*

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:

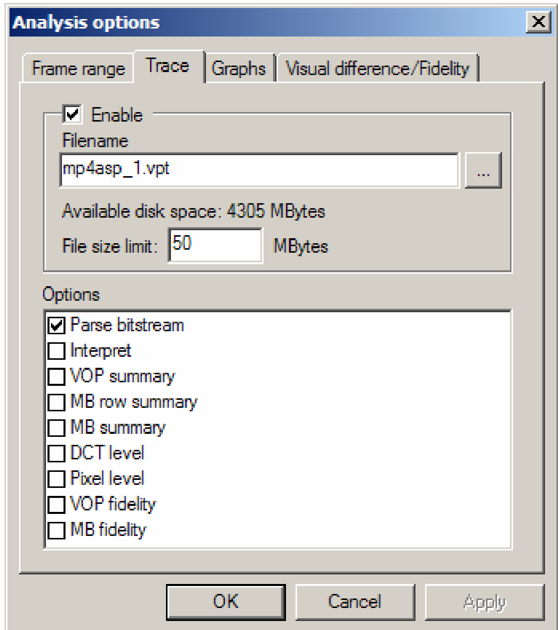
- hex (i.e. hexadecimal) view of stream data
- the structure of the file being analyzed
- analysis of the video buffer usage.

The 'Analysis options' menu item provides access to all the Trace, Graph and Fidelity options as well as setting of the range of frames over which to collect the Trace, Graph and Fidelity information.

NOTE. *The 'Trace enable..', 'Graph enable...', 'Fidelity enable...' and 'Analysis options...' cannot be set in pause mode - the video MUST be stopped.*

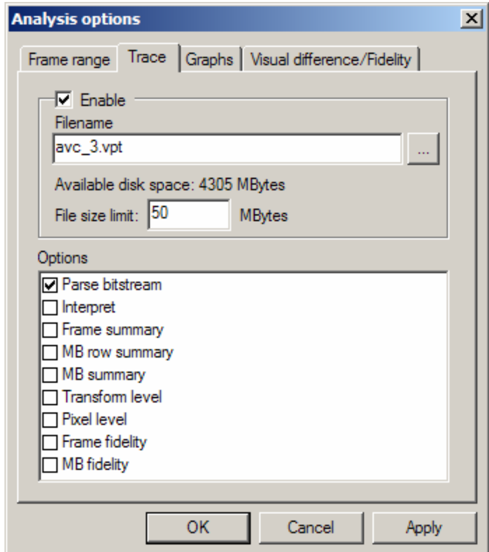
This is because the MTS4EA could not otherwise verify that the Graph and Fidelity data were collected over the correct range of frames. For example, if the video were paused at frame 23 and Trace were enabled for a range of frames from 20-25, the Trace data would be incorrect.)

Trace enable Ctrl+T

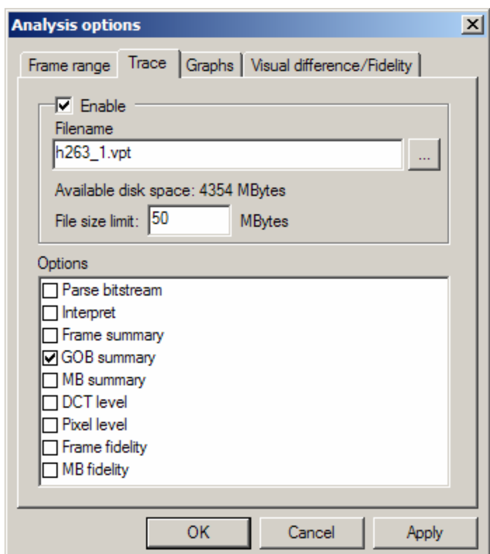


The Trace options provide text outputs which describe the contents of an encoded video stream, to various levels of detail.

The Trace options available vary with each video standard: the screenshot above is for MPEG-4; the screenshots below for H.264/AVC and H.263 respectively. (Unless otherwise stated, each option applies to every video standard.)



H.264/AVC



H.263

NOTE. *Using these options can generate a lot of data - files hundreds of MBytes 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).*

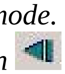
Alternatively, the user can set a maximum file size, after which no more trace data is output to the file concerned - see Trace enable Ctrl+T, page 7-102.

HINT. *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.*

NOTE. *To set the range of frames over which to collect the Trace data, click on the 'Frame range' tab.* *See Trace enable Ctrl+T, page 7-102.*

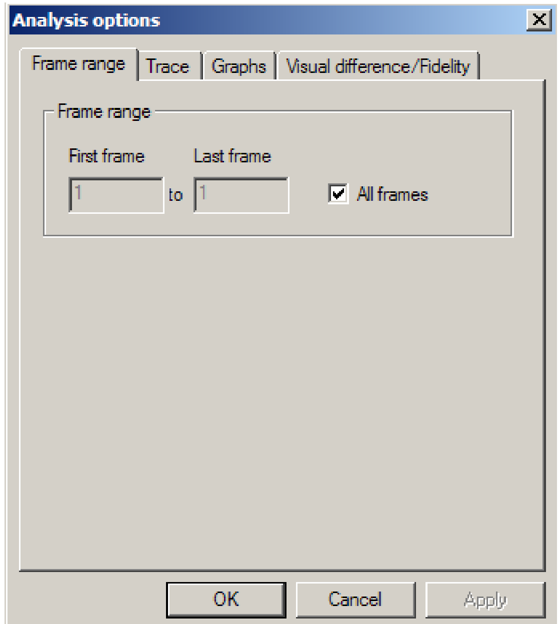
Frame Range tab. This option allows the user to specify the start and end frames between which to gather the Trace and Graphs information.

NOTE. *The frame range set on this tab is the same frame range as used for the collection of MacroBlock statistics, where these are collected over a range of frames* *(see MB statistics..., page 7-83).*

NOTE. *'Frame range...' is unavailable (grayed out) in pause mode. (Indicated when the Pause/Step forward icon*  *or Pause/Step backward icon*  *is pushed in on the toolbar and 'Play' menu.) This is because the frame range over which statistics are to be accumulated cannot be changed during pause mode.*

This is because the MTS4EA could not otherwise verify that the Trace and Graph data were collected over the correct range of frames. For example, if the video were paused at frame 23 and Trace were 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:

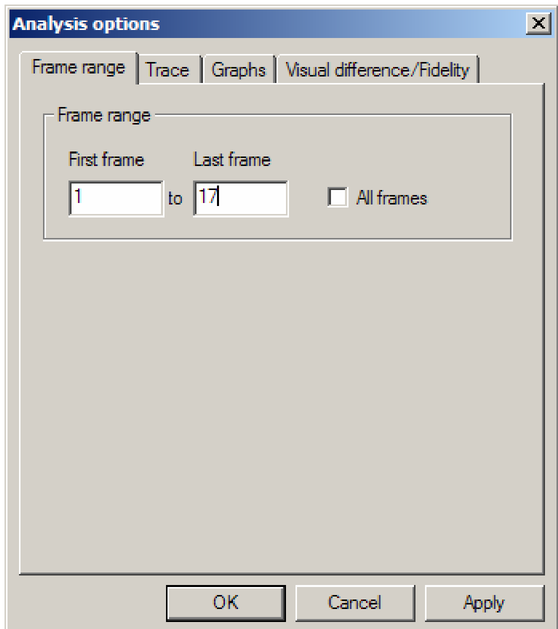


All frames

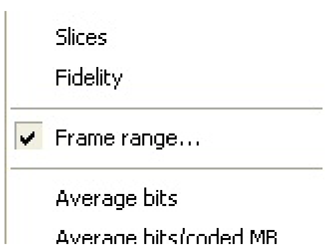
This option is the default and calculates the sequence statistics from the start of the sequence until the current frame.

To set a range of frames

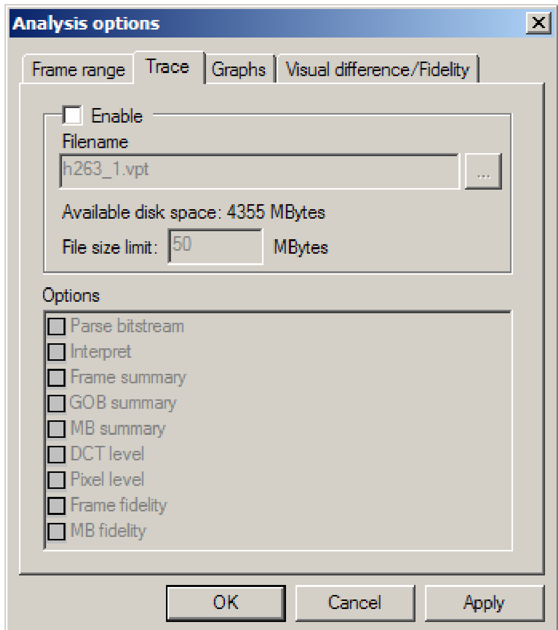
If the 'All frames' check box is cleared, then a range of frames can be entered (in this case, frames 1 to 17 inclusive):



NOTE. *When a range of frames has been set, the 'Frame range...' item on the MB statistics menu changes, with a tick to indicate a range of frames has been set:*



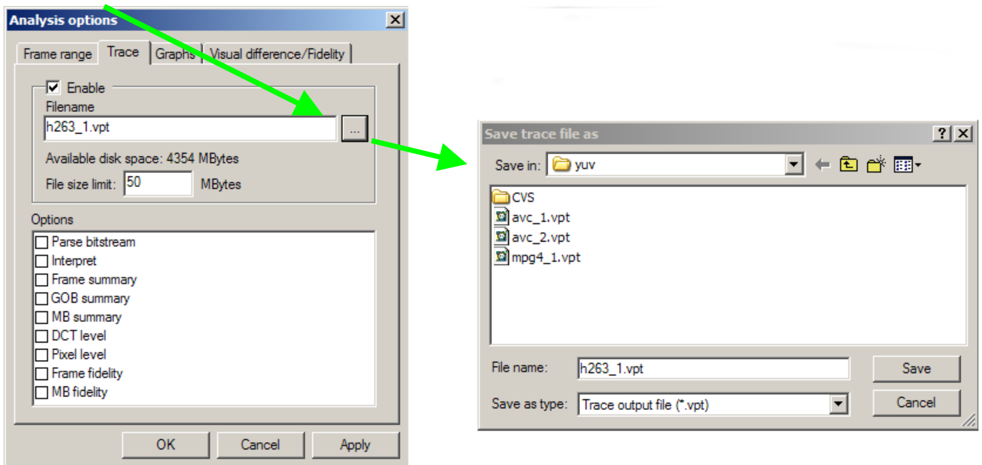
Enable. This option Enables/Disables the Trace output. If the output is disabled, the remaining selections on the Trace tab are unavailable (grayed out) although the values are stored during the current use of MTS4EA. (They are reset to their default values the next time MTS4EA is started.)



Filename. By default, the filename suggested is the same as the base name of the input video file (i.e. 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')

You can also browse to a specific folder and enter the filename there, by clicking on the [...] button:



File size limit (Available disk space). Trace files can generate a lot of data: for example, 2-3 Mbytes 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 MBytes in size.

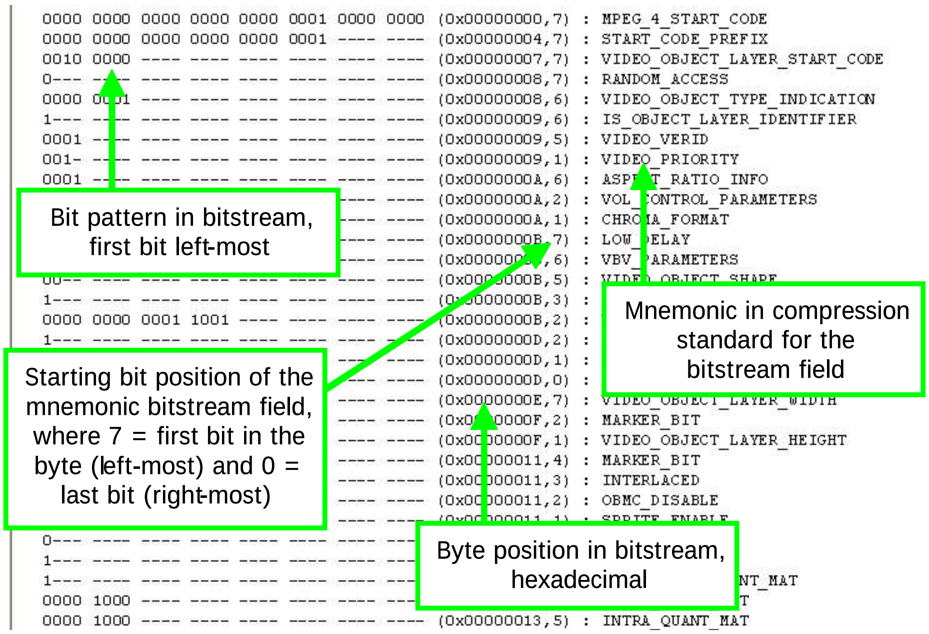
This entry allows the user to limit the amount of disk space taken by the Trace file.

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

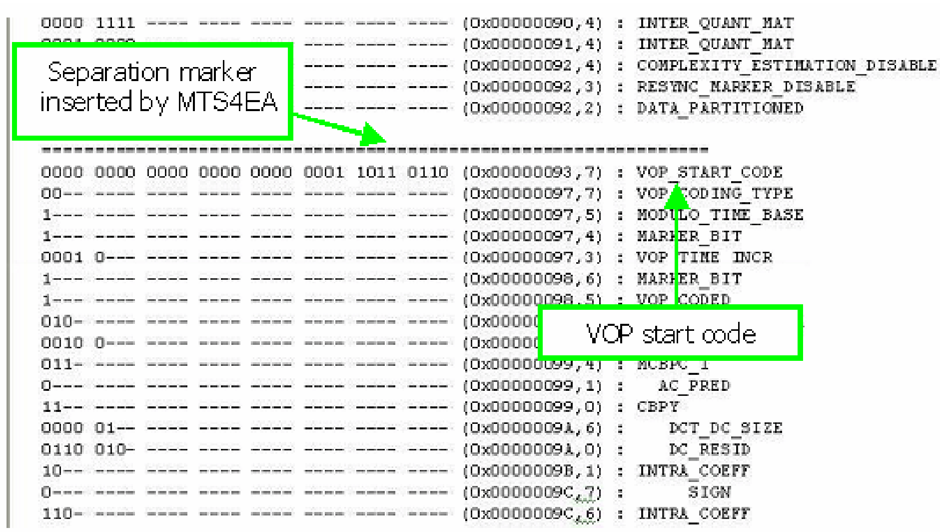
Parse Bitstream. If this option is selected, then the bitstream is 'parsed' to a file. This gives the individual bit patterns for the bitstream fields as well as the mnemonic used within the standard to identify the field.

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

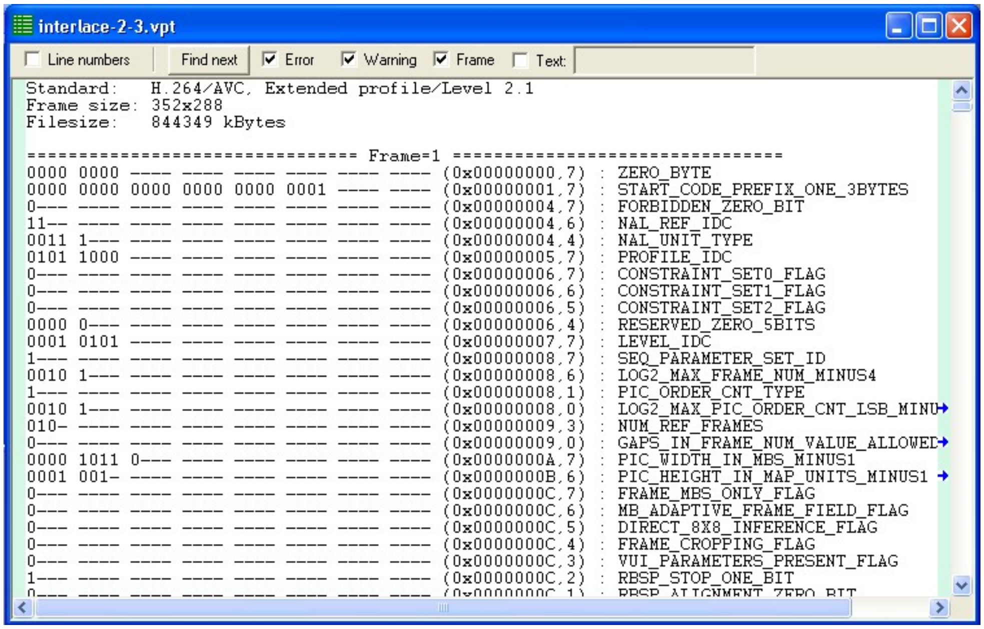
NOTE. *See Bitstream Syntax Debugging using MTS4EA, page 8-9 for detailed information on the meaning of these.*



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)

```
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

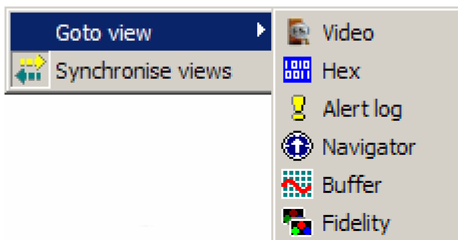
NOTE. *Also check out Interpret (see Trace enable Ctrl+T, page 7-102) as another excellent tool to do bitstream syntax debugging.*

HINT. *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 and Parse Bitstream only) which allows:

- finding the next instance of the words 'Error', 'Warning' or 'Frame', or the text entered
- going to the area in another view which corresponds to the area which has been selected in the Trace file:

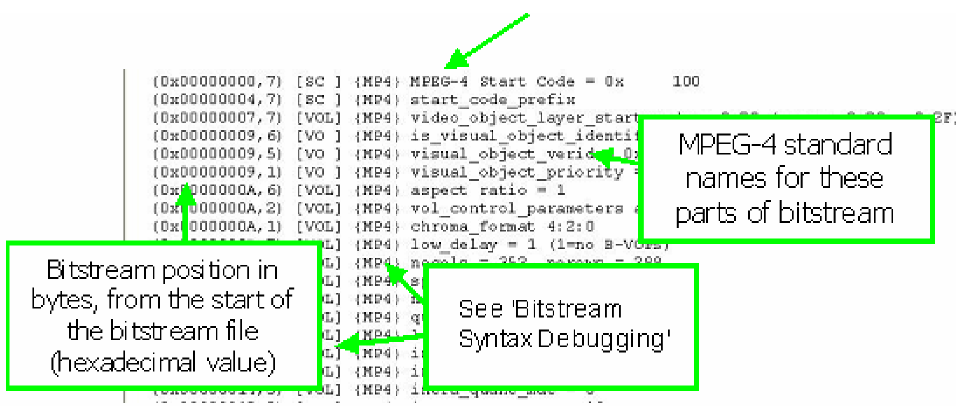


See Synchronized views/navigating the views, page 7-6 for more information.

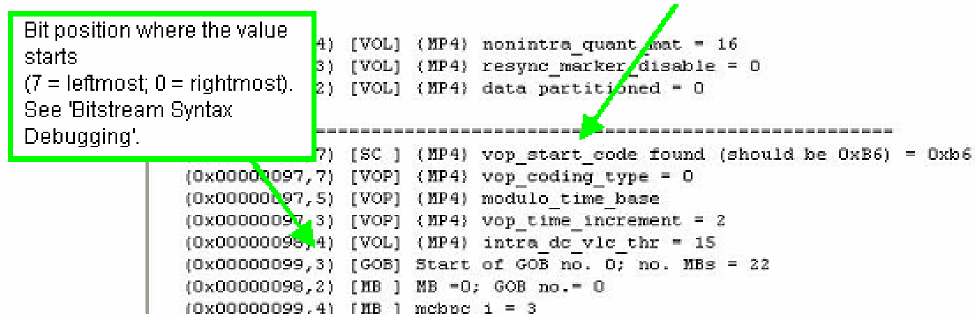
NOTE. *This right-click context-sensitive menu is only active in the Trace/Interpret and Trace/Parse Bitstream trace files.*

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

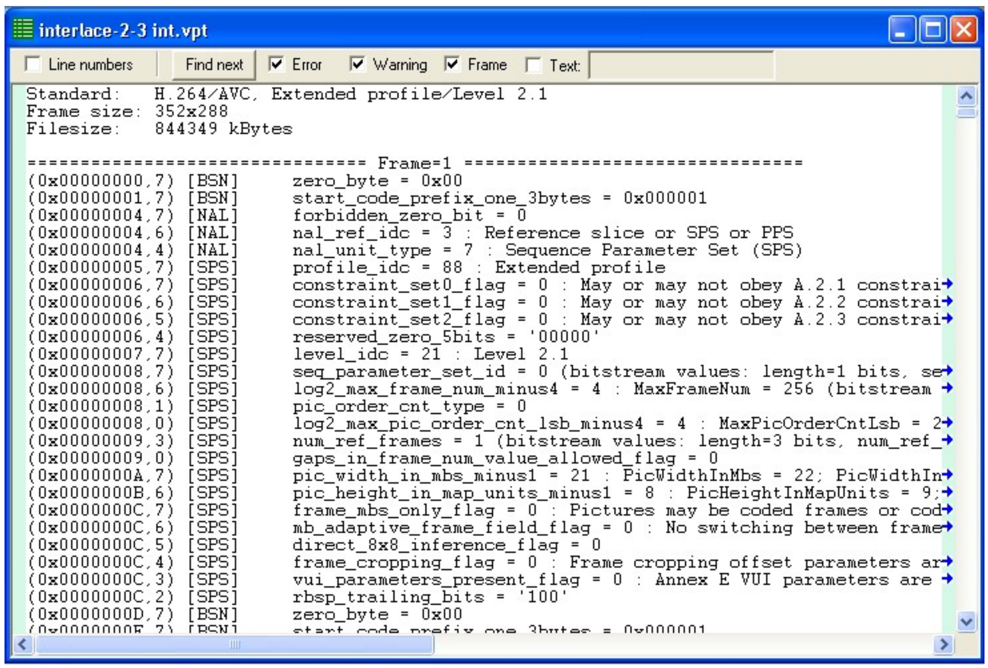
NOTE. *It is important to also see General codes used in Trace Files and Alerts, page 8-1 for an explanation of some of these items.*



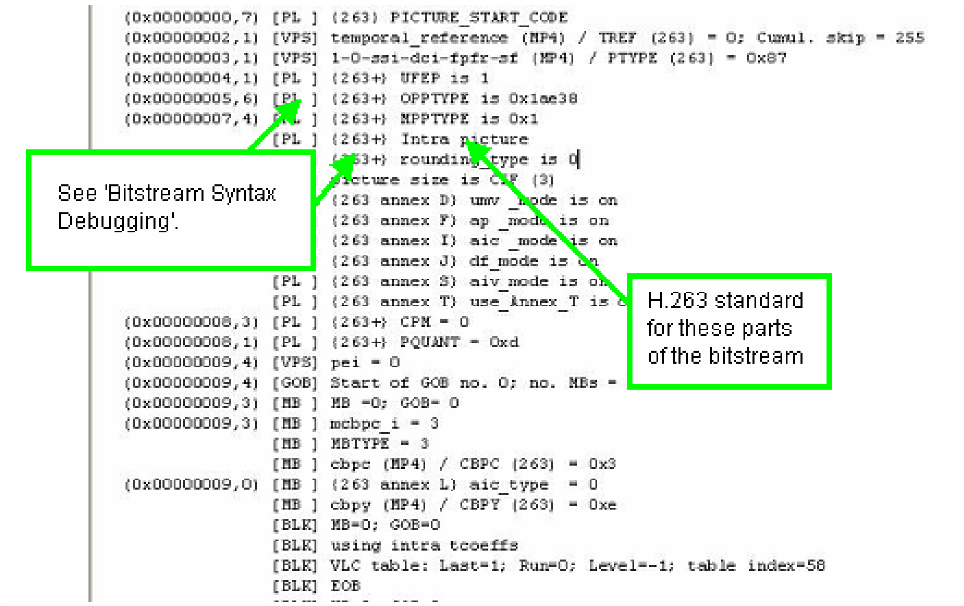
MPEG-4 Example, at start of bitstream



MPEG-4 Example, at start of VOP



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



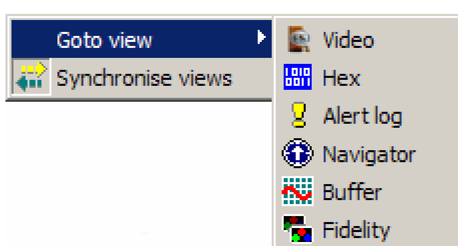
H.263 Example at start of bitstream

HINT. *To quickly find if there are any errors in the Trace file, simply use the 'Find next' function of 'View trace...', looking for the 'Error' and 'Warning'.*

<u>Go to other views/right-click menu</u>

By right-clicking with the mouse, a context-sensitive menu appears (Interpret and Parse Bitstream only) which allows:

- finding the next instance of the words 'Error', 'Warning' or 'Frame', or the text entered;
- go to the area in another view which corresponds to the area which has been selected in the Trace file:

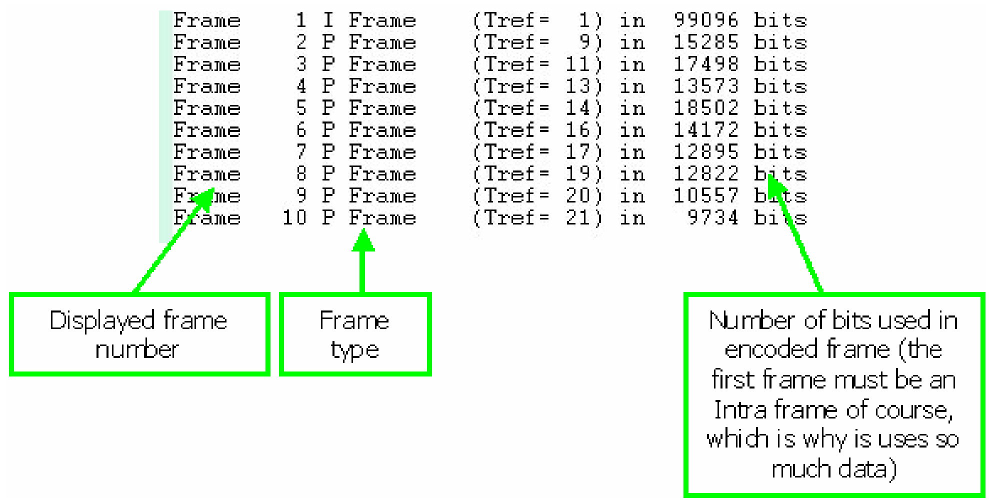


See Synchronized views/navigating the views, page 7-6 for more information.

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

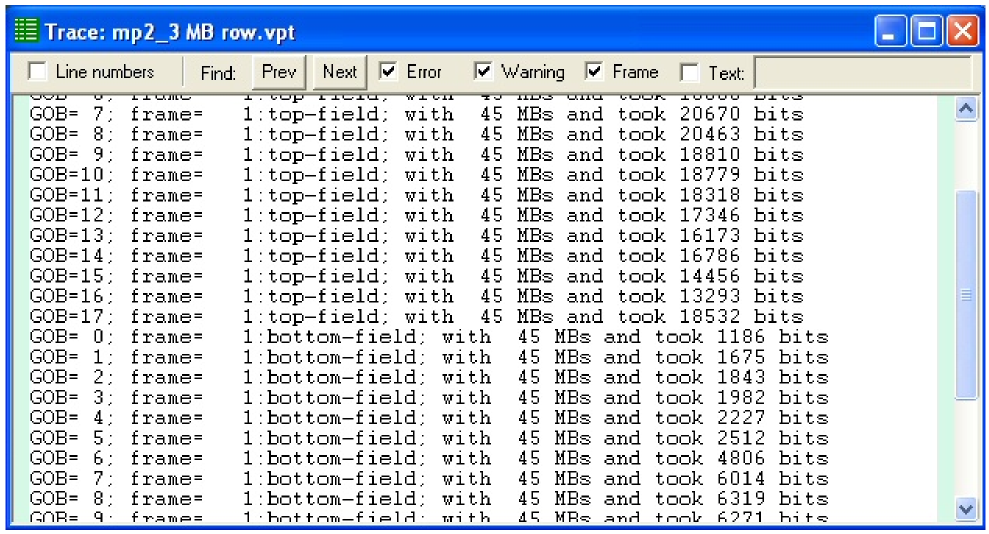
Frame Summary [H.264/AVC, MPEG-2, H.263 and H.261] or VOP Summary [MPEG-4] . If this level of Trace is selected, then the Frame summary is output to the file, specifying the number of bits generated, the index of the decoded frame, the frame type, the temporal reference (if appropriate) and other data.

(this example is for H.263, where 'Tref' is used (see Summary Tooltip Ctrl+U, page 7-57).

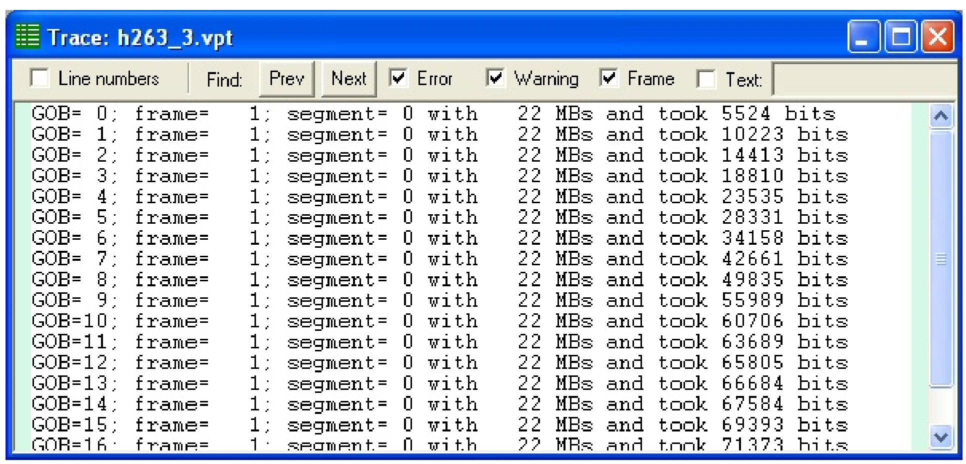


MB row summary [H.264/AVC, MPEG-4, MPEG-2] or GOB Summary [H.263, H.261]. This Trace outputs summary data for each horizontal row of MacroBlocks, starting from the top of the frame and working downwards.

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)



H.263 Example

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
```

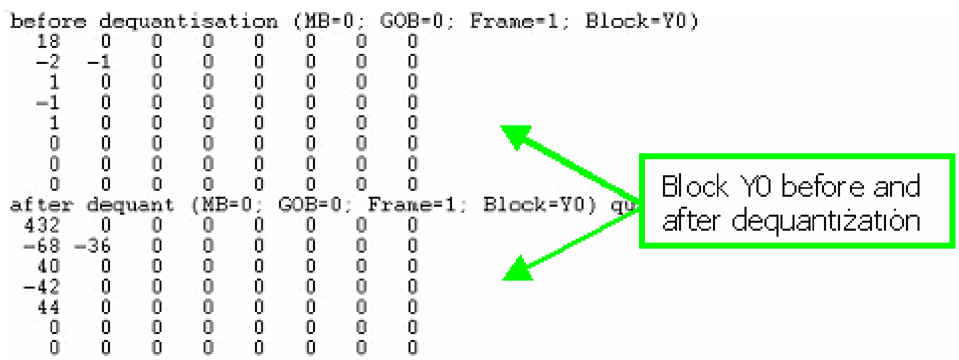
MPEG-4 Example

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 actually of transform coefficients which are not actually DCTs.*

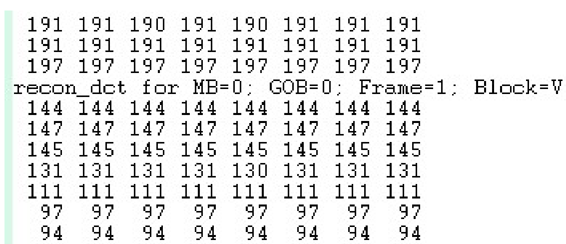
For MPEG-4, H263 and H.261 this gives 3 sets of information:

- '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;
- 'recon_dct' = the inverse DCT residual values, before summing with any prediction.



```
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) qu
 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
```


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:

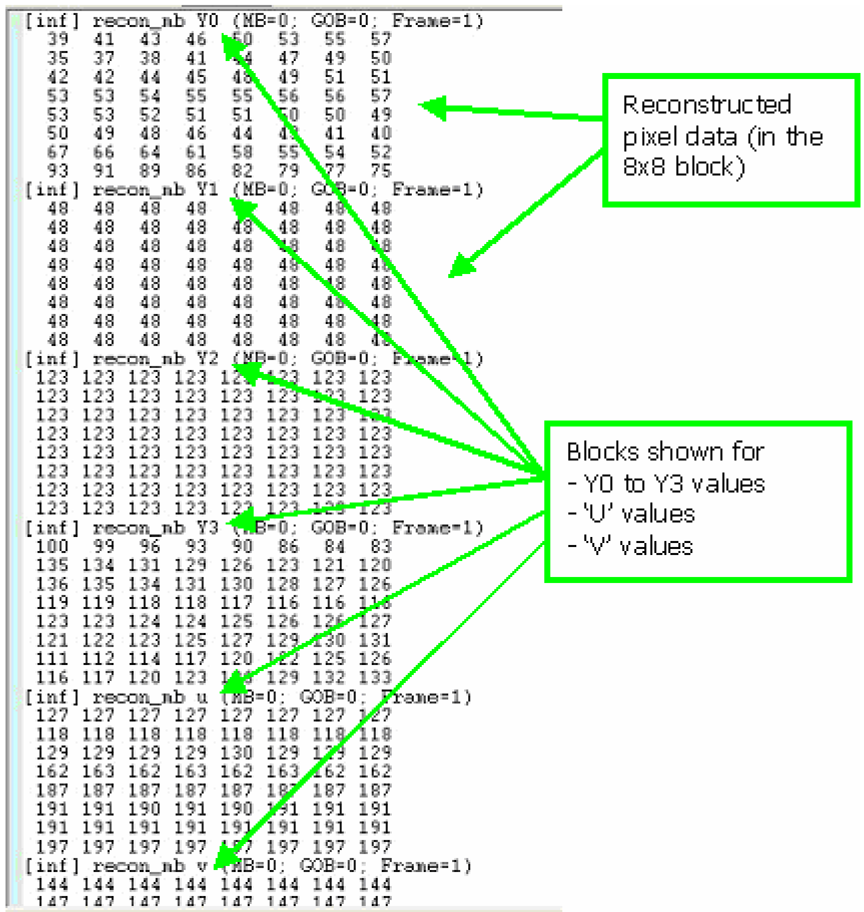
```
levels (MB: 0; Frame: 1; Block 0:Y0)
 -2816  -640  -256  -320
     0     0     0     0
     0     0     0     0
     0     0     0     0
levels (MB: 0; Frame: 1; Block 0:Y1)
  -256     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
levels (MB: 0; Frame: 1; 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
```

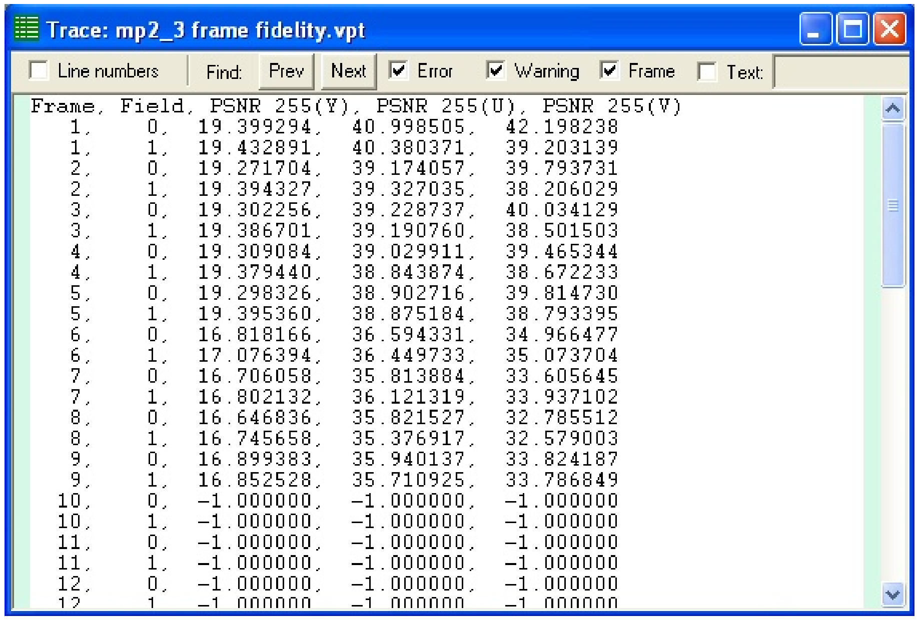
NOTE. *This can generate a lot of data.*

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.*

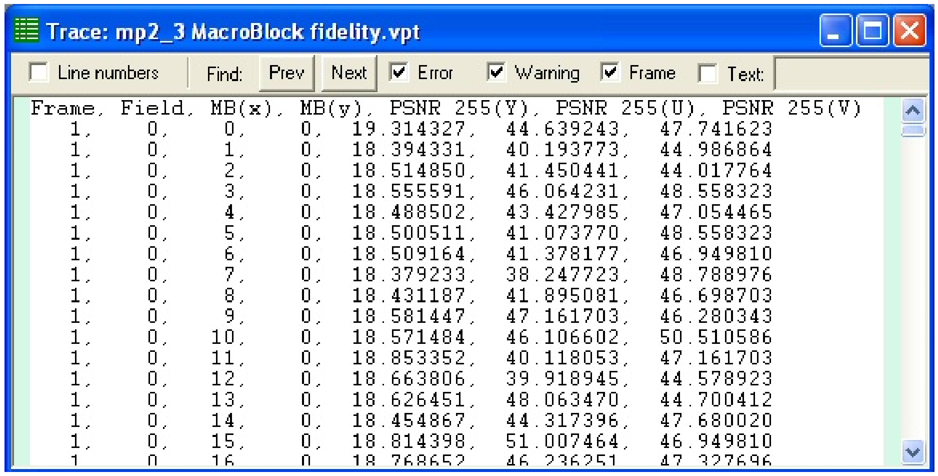
Frame fidelity [H.264/AVC, MPEG-2, H.263 and H.261] or VOP fidelity [MPEG-4]. This provides a Trace output of the fidelity analysis frame-by-frame. See Fidelity enable..., page 7-132 for more information on fidelity analysis.



NOTE. *The lines in the screenshot above for frames 10 onwards are empty because the corresponding YUV file stops at this point (there are no more frames in the YUV file).*

HINT. *Some of the example files provided have the necessary corresponding YUV files for fidelity analysis - see Example files..., page 7-31 for more information.*

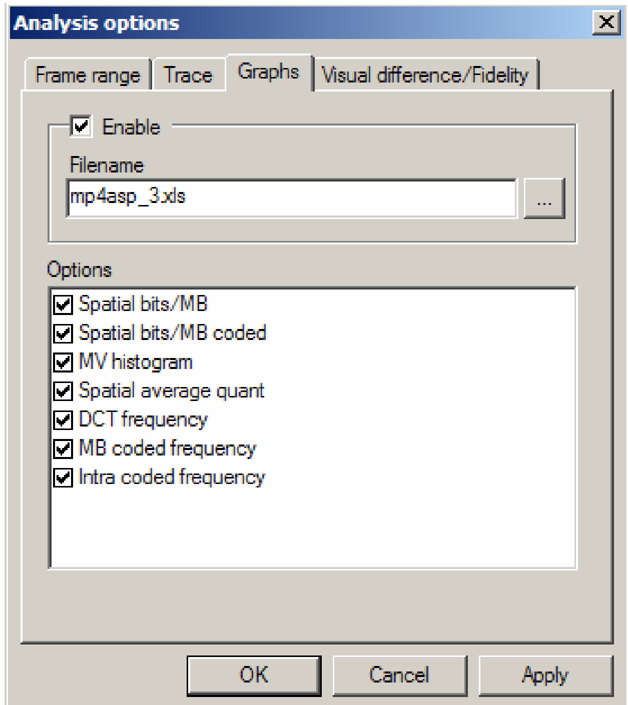
MacroBlock fidelity. This provides a Trace output of the fidelity analysis for each MacroBlock in the frame range. See *Fidelity enable..., page 7-132* for more information on fidelity analysis.



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

HINT. *Some of the example files provided have the necessary corresponding YUV files for fidelity analysis - see Example files..., page 7-31 for more information.*

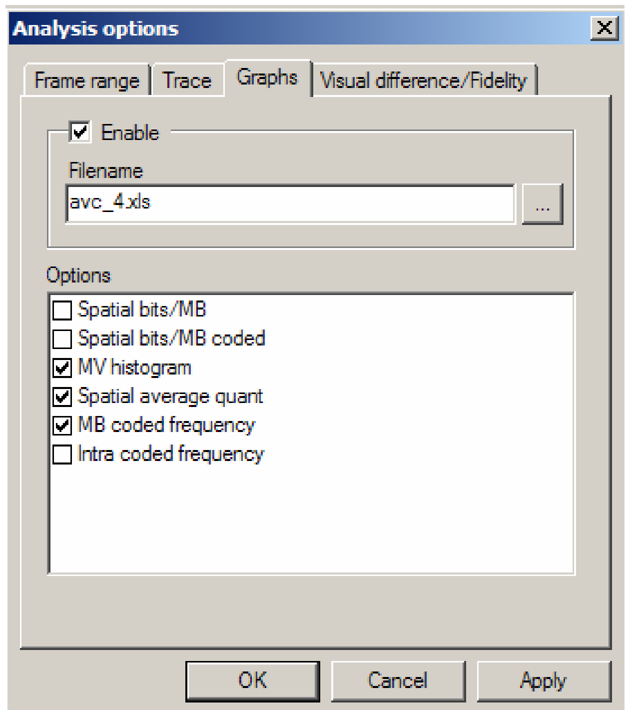
Graph enable... Ctrl+G



'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):



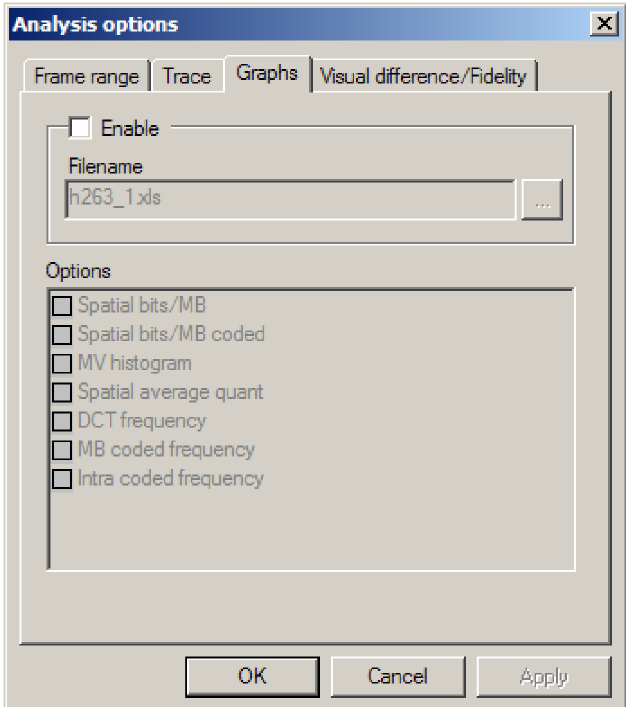
Many of these graphs are equivalent to the statistical summaries that can be overlaid on the final frame, but are presented in a more visual format.

NOTE. *MTS4EA assumes that the program Microsoft Excel is used for analysis of the statistics provided, 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 *Trace enable Ctrl+T, page 7-102*.

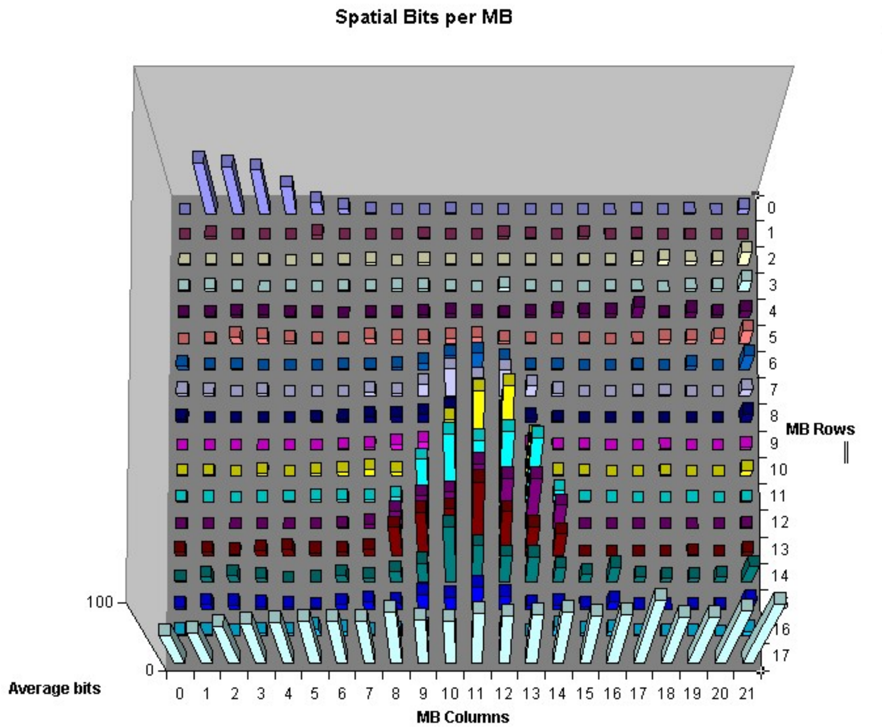
Enable. This option enables/disables the collection of the statistics for the Graph data. When disabled, the Graph tab appears as below:



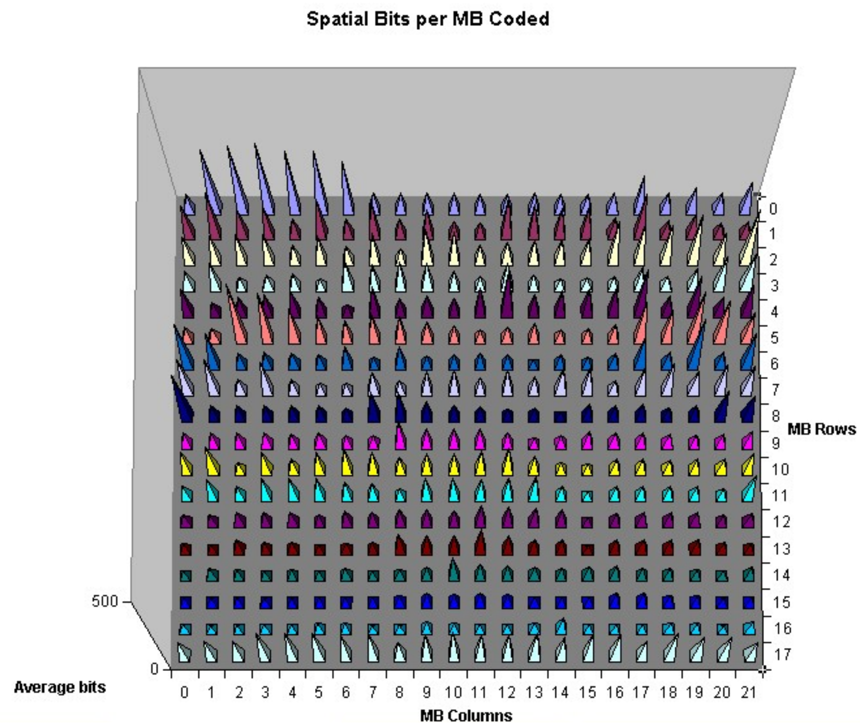
Filename. By default, the filename suggested is the same as the base name of the input video file given the extension '`.xls`'.

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 '...'

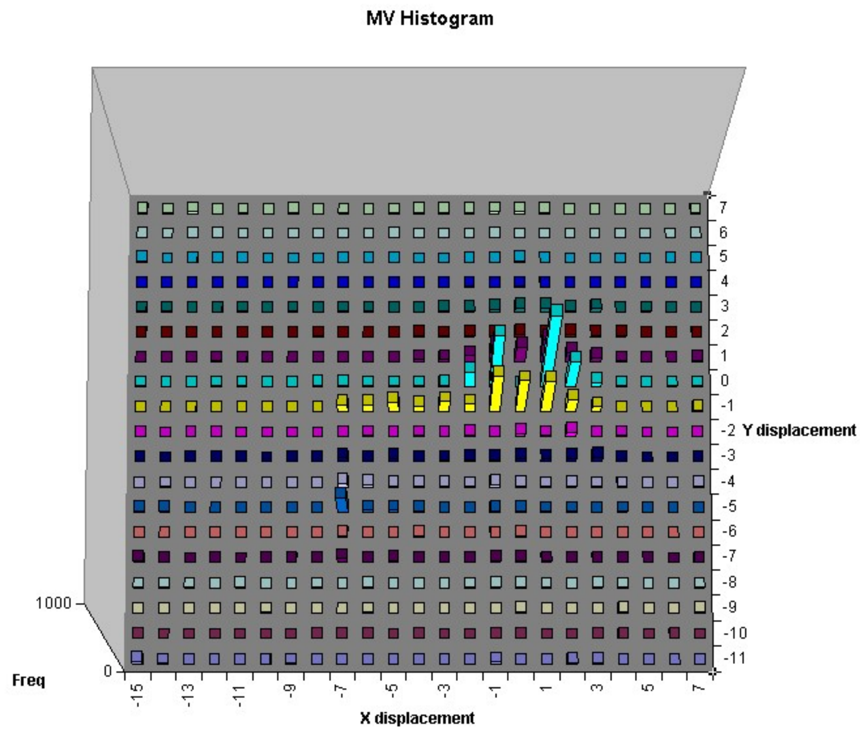
Spatial bits/MB. This option creates a graph which shows the Average Bits per MacroBlock as a contour plot, over a 2-dimensional plane that represents the picture. The contours are plotted in different colors, and the meanings of the colors are represented in a key positioned at the side of the graph.



Spatial bits/MB coded. This option creates a 3-dimensional pyramidal graph, where the height of the pyramid represents the average number of bits each MacroBlock used for each time it was coded.

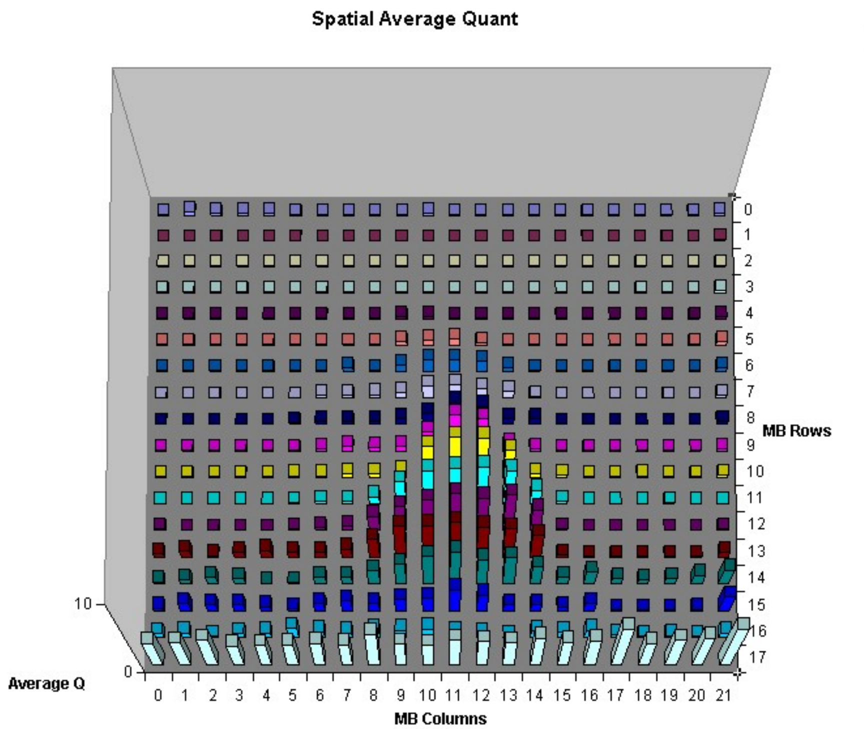


MV histogram. The motion vector histogram creates a graphical representation of the frequency of use of different motion vectors. This can be used to identify the range of search of the motion estimator in the encoder. The heights of the blocks represent the frequency with which the motion vectors have been chosen.

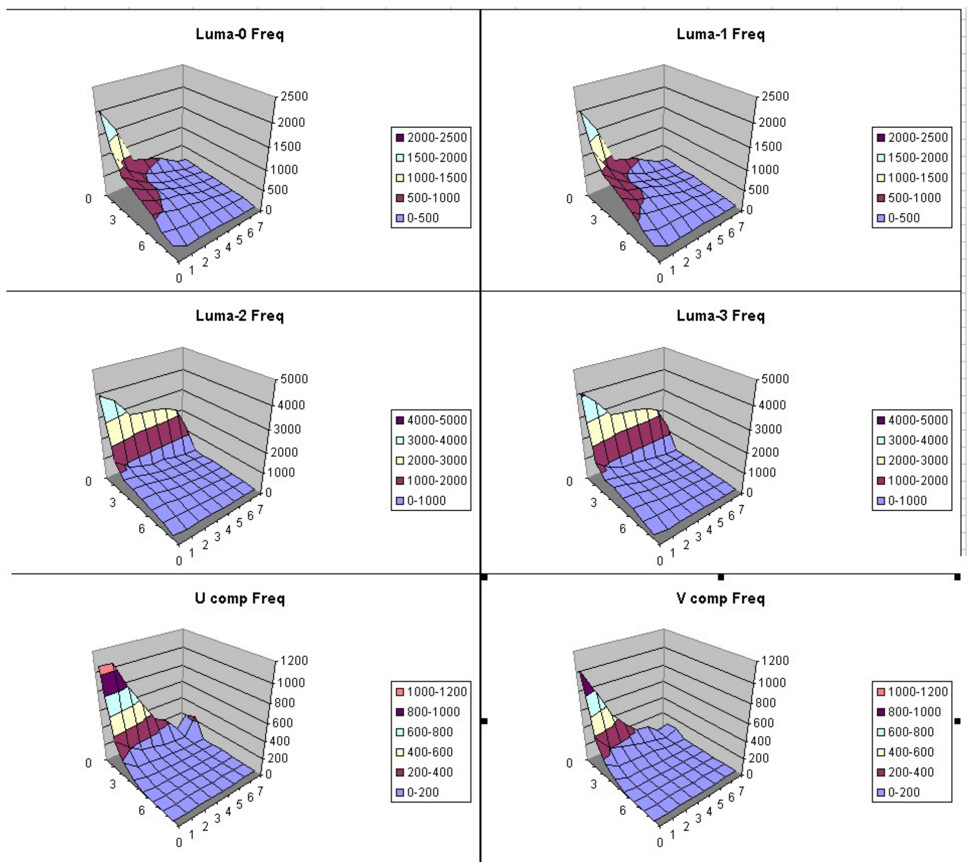


NOTE. *An Excel spreadsheet can only be 256 columns wide, so if the motion vectors are outside the range +/- 127 then the values are put into 'bins' to scale the values within the 256 columns. This is done by dividing the values by factors of 2 until they fit within 256 columns. In this situation, the correct range is shown on the axes.*

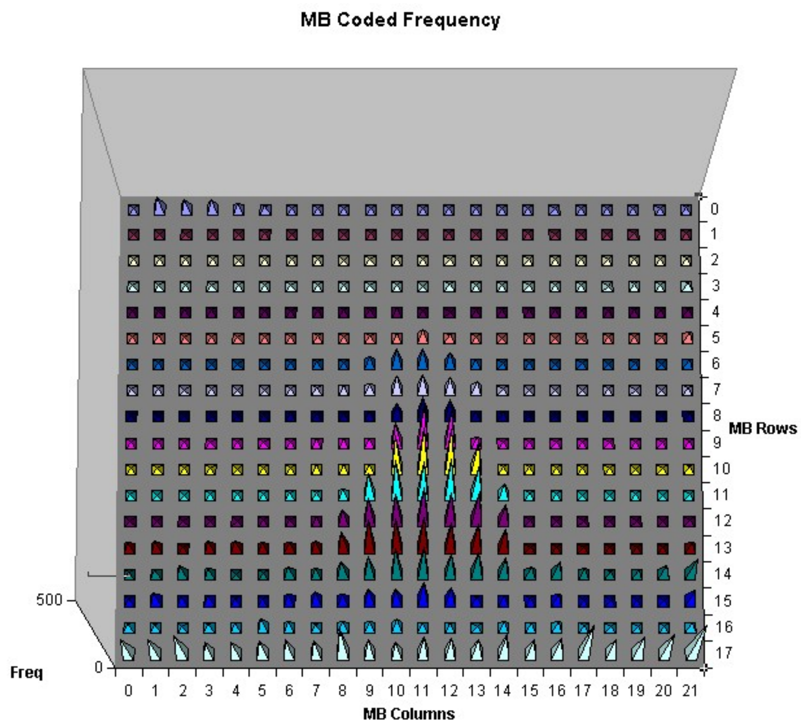
Spatial average quant. This option creates a graph of the quantizer choice at each part of the picture. This can be useful for determining poor quantizer control in certain encoders (often too low quantizers are chosen at the top of pictures, which then overshoots the bit-budget and then higher quantizers must be used progressively as the frame is encoded - meaning that the top of the picture is always coded more accurately than the bottom).



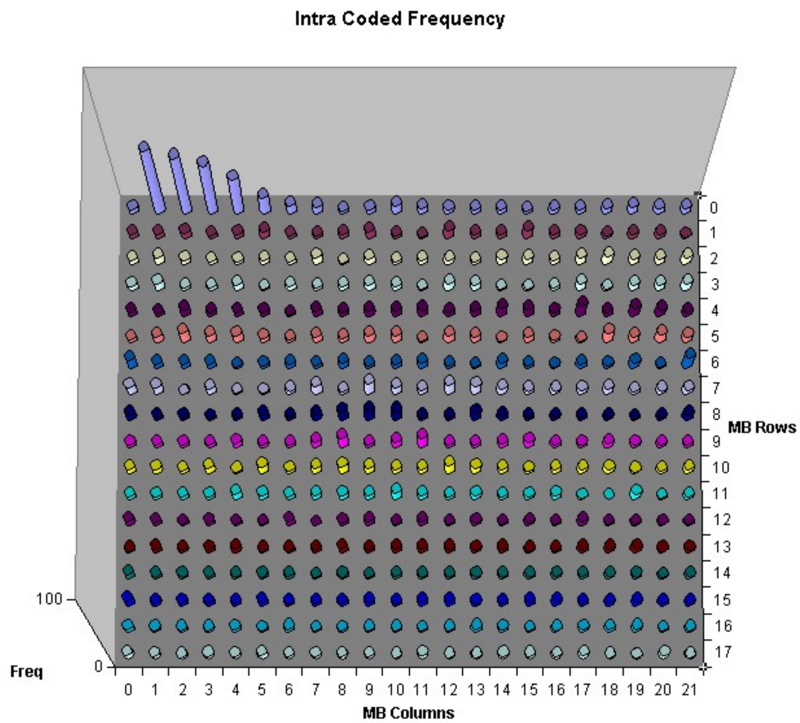
DCT frequency [MPEG-4, MPEG-2, H.263 and H.261 only]. This option creates a graph of the frequency with which the different components of the Discrete Transform Cosine are used. The positions of the transmitted coefficients are recorded and shown here for each of the six 8x8 blocks that comprize the 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.

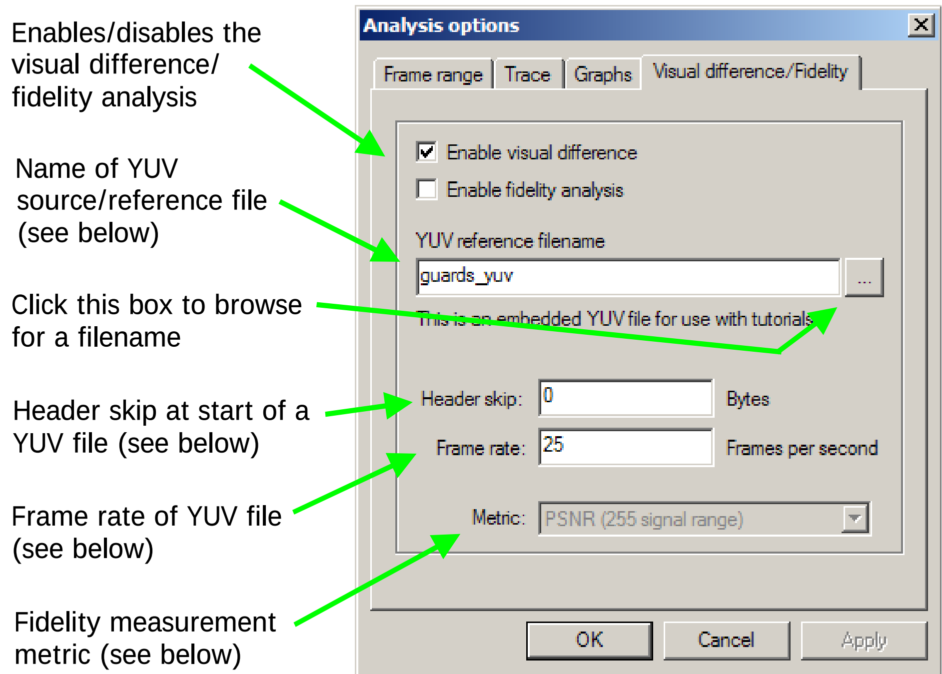


Intra coded frequency. This option creates a graph of the frequency of intra coding of each MacroBlock. Over time, each MacroBlock may be updated in Intra mode every so often (as required in the Standards) in case there is a line error, or a mismatch between encoder and decoder. Encoders which do this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



Fidelity enable...

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, page 7-102*;
- as real-time overlays, when the overlay is selected - see *MB statistics..., page 7- 83*.

YUV reference filename. The name of the file used as the reference or source of encoding, of the encoded stream.

This file must be in the following format:

- one byte per sample
- progressive scan (not interlaced)
- row raster order (top picture row first)

- planar YUV 4:2:0 sub-sampled i.e. 4 bytes of Y data for each byte of U data and each byte of Y data
- all the Y plane values for a frame are together, followed by the U values for a frame, followed by the V values for a frame
- Y plane values are 0-255 unsigned
- U and V plane values are unsigned with a DC offset of 128.

HINT. *There are built-in example YUV reference files for 3 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 filenames are ***automatically*** *filled in. See Example files..., page 7-31 for more information.*

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).

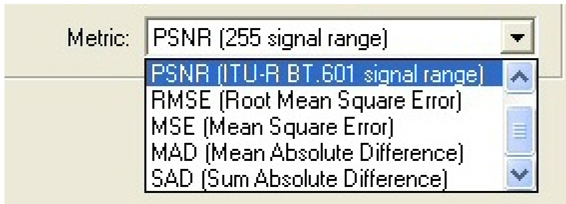
Frame rate. The rate at which to the YUV frames were recorded, in frames per second.

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 to 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 YUV frame to associate to which encoded frame; MTS4EA works out the time code for each YUV frame based upon the Frame Rate value and associates the YUV 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 YUV domain (i.e. independently for Y, U and V).

The fidelity metrics are calculated per MacroBlock; the frame values represent the average value for all the MacroBlocks in the frame.

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

| … |	denotes taking the absolute value of an expression
Σ	denotes the summation of an expression over the range of (x, y) in the image plane
f(x,y)	is the sample value at the location (x, y) in the reference image plane
g(x,y)	is the sample value at the location (x, y) in the decoded image plane
N	is the total number of samples in the image plane

PSNR (255 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.

$PSNR = 10 \cdot \log (\hat{S}^2 / RMS^2)$

For PSNR (255 signal range) the peak image value is assumed to be 255 in all three image planes (i.e. this applies for Y, U and V planes):

$PSNR_{255} = 20 \cdot \log (255 / RMS)$

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.

$RMS = \sqrt{MSE}$

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)

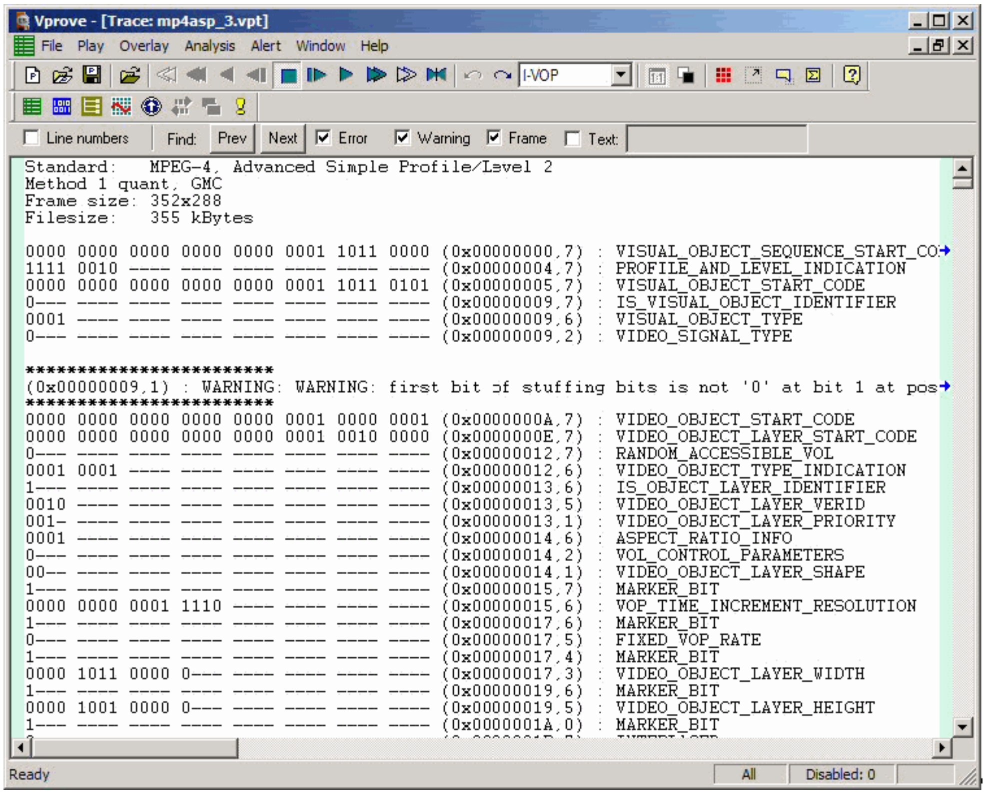
The SAD metric is defined as the sum of absolute differences between samples in the reference and decoded image planes.

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

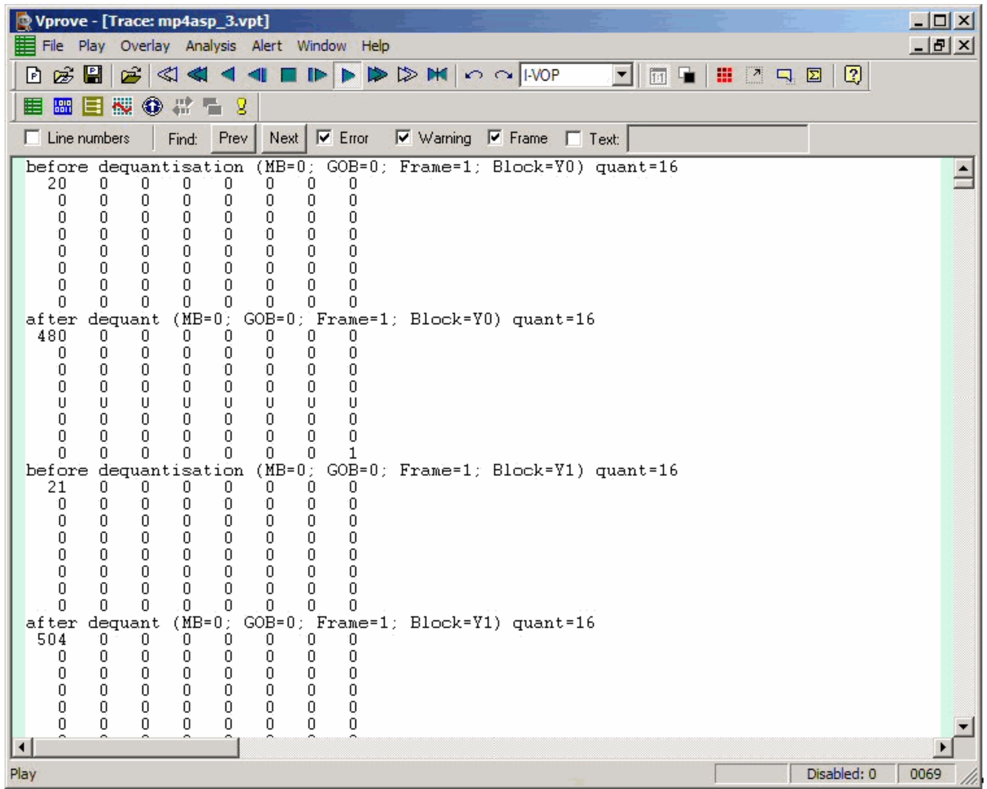
View trace... Ctrl+V

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:



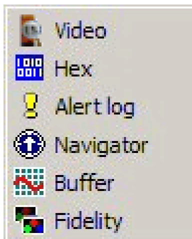
NOTE. If there is not a 'current' Trace file (and 'Trace enable' is not enabled) then this menu option/toolbar icon is unavailable, i.e. grayed out.

HINT. *Multiple 'View trace...' windows can be opened at the same time.*

HINT. *If the video stream has been rerun and a Trace file saved with the same name as that already open, then clicking the 'View trace...' menu option or icon or Ctrl+V will refresh the currently open Trace file.*

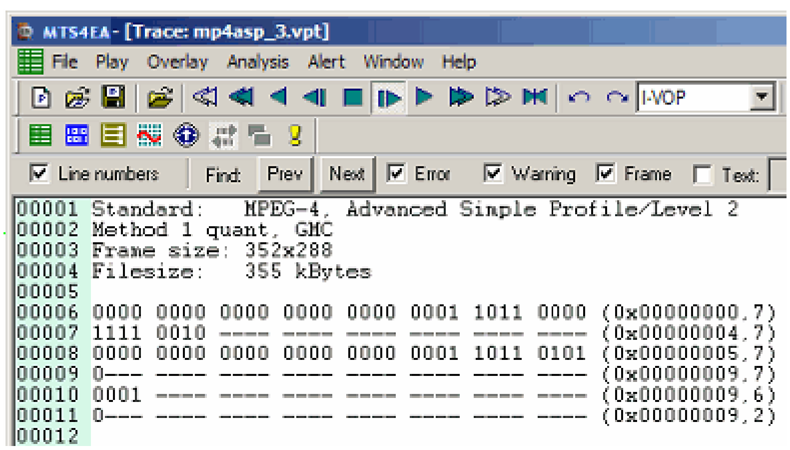
Go to other views/right click menu - Interpret and Parse Bitstream only. By right-clicking with the mouse, a context-sensitive menu appears which allows:

- finding the next instance of the words 'Error', 'Warning' or 'Frame', or the text entered;
- 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*

Line numbers. Line numbers can be switched on/off by enabling or disabling the 'Line numbers' checkbox:



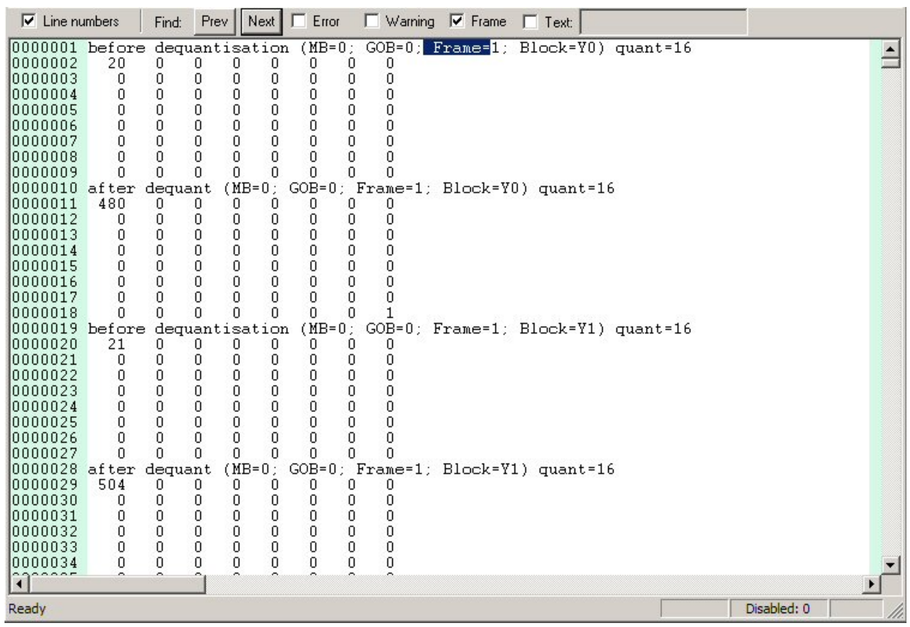
Find data. The Trace file can be searched for any data, using the Find: 'Prev' (i.e. previous) and 'Next' buttons. This finds the previous/next occurrence of any of the enabled strings.

***HINT**. The 'F3' key can also be pressed, to 'find next'.Similarly, the 'Shift+F3' key can also be pressed, to 'find previous.*

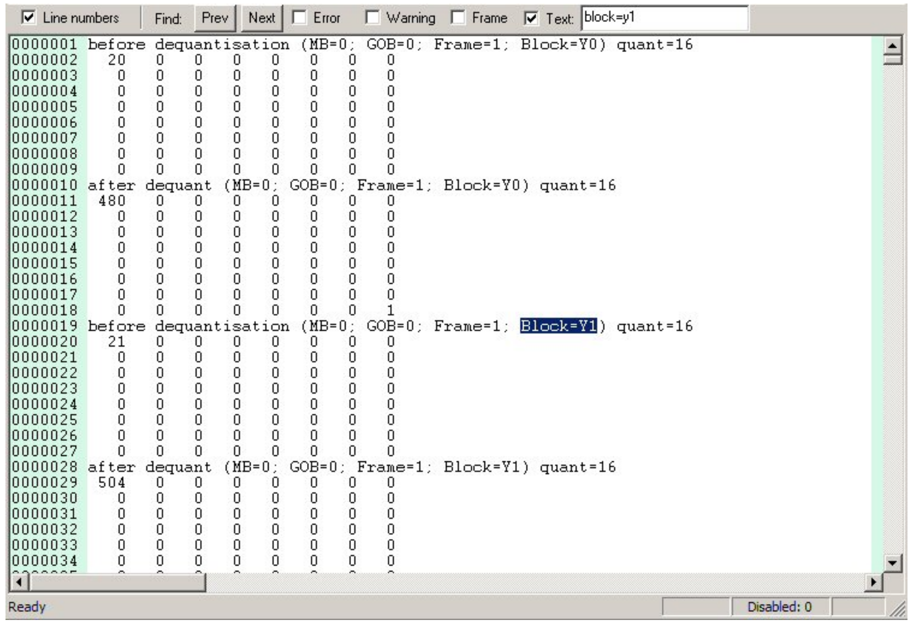
There are some standard strings that are useful to search for:

- 'Error'
- 'Warning'
- 'Frame'.

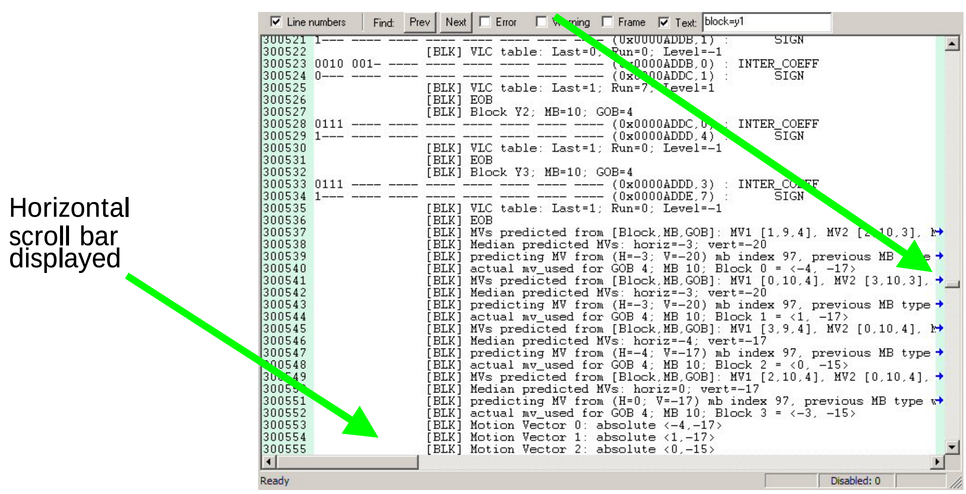
These are found by enabling the relevant check box ('Frame' example below):



To search for any text, select the box next to 'Text' and enter the text in the box (in this example, searching for block=y1'):



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;
- as it is intended in future that this will contain other data.

View Graphs

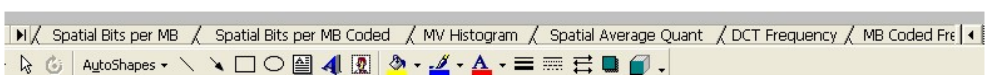
This option calls Microsoft Excel to display the graphs generated: a macro is then run within Excel to create the graphs.

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

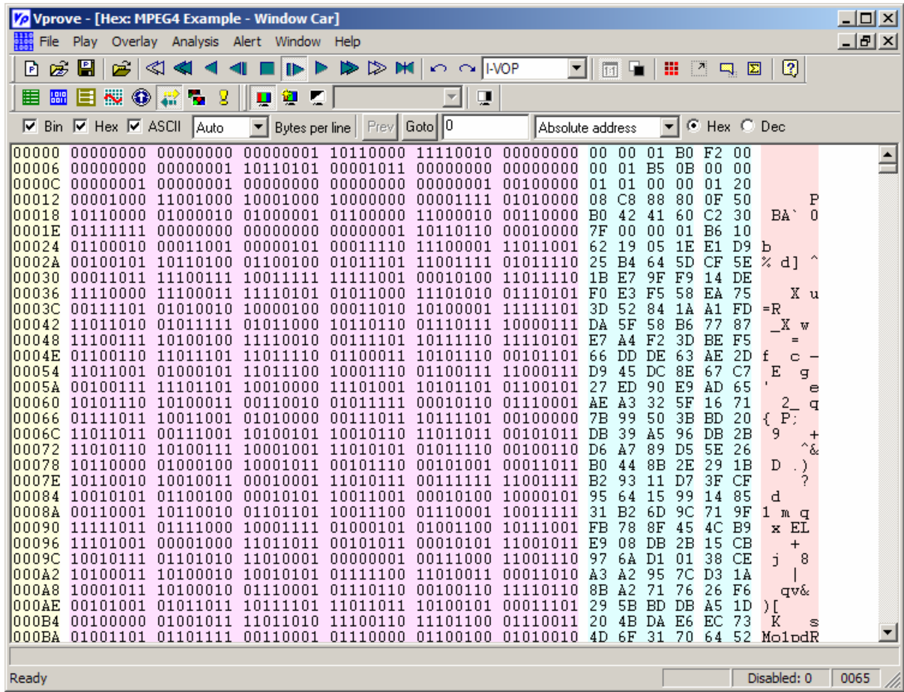
The default selection within Microsoft Excel is (usually) to disable macros; you must click on 'Enable Macros' if a warning dialog is displayed.

HINT. *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).

HINT. *Multiple 'View hex...' windows can be opened at the same time.*

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

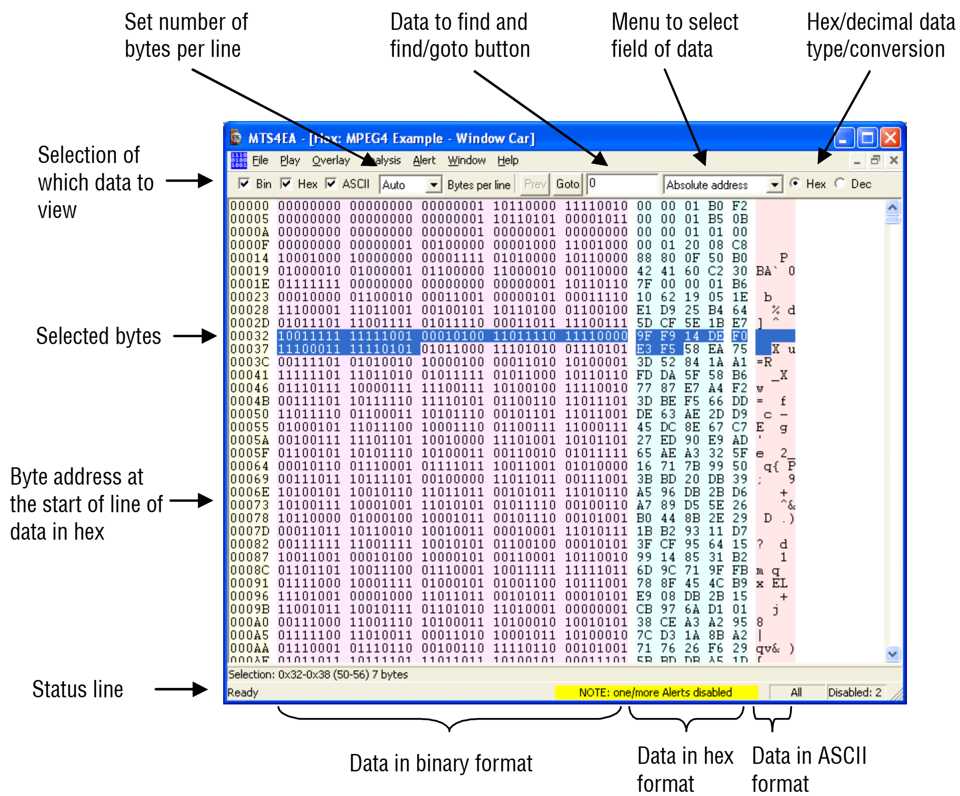
- absolute address (that is, from the start of the file);
- relative address (that is, from the currently-selected location;
- bit patterns;
- hex data;
- ASCII data.

HINT. *In the field where the data to find is entered, a wildcard character can be entered - this is '.' (a period).*

HINT. *Also, entering a hex value, then clicking 'Dec' will convert this number to decimal (and vice-versa).*

(See below for information on how to use these features.)

Sections of HexView window.



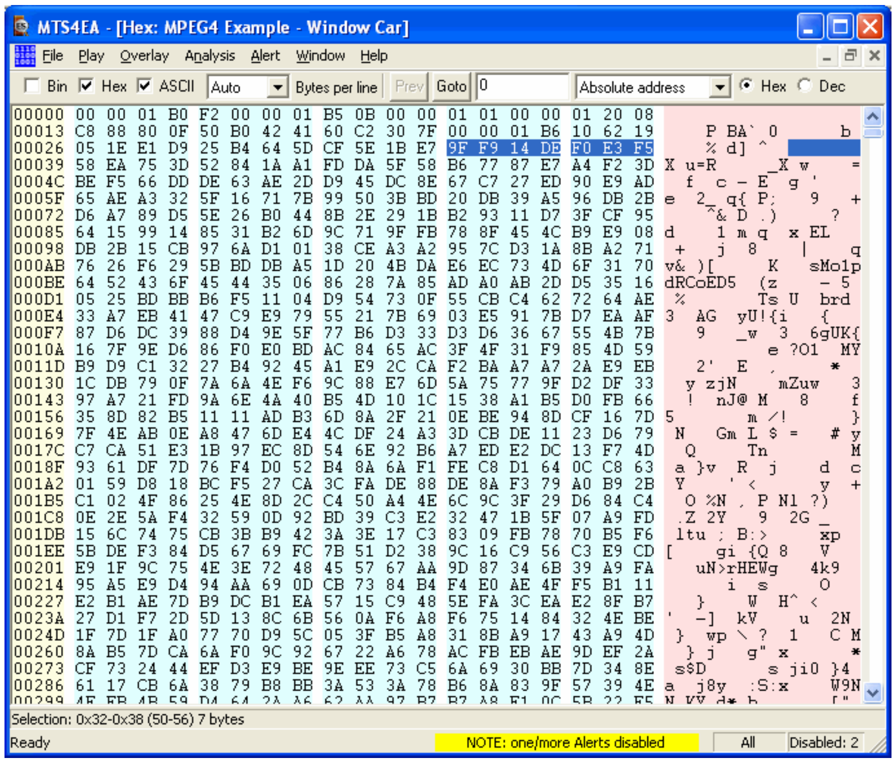
Setting information displayed; window width. The width, size etc. of the HexView window can be set as with any other window in Windows.

The same data is shown in each of the three sections:

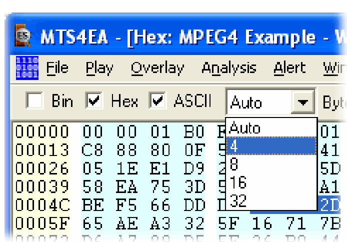
- binary
- hexadecimal
- ASCII

in their respective formats.

Each of these areas can be individually displayed/not displayed by clicking the appropriate check box in the top left. In the example below, the binary display has been 'switched off', leaving only hex and ASCII.

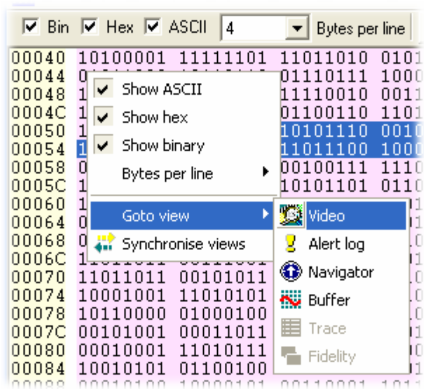


Setting bytes per line. The menu is used to set the number of bytes per line:



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.

Right-click pop-up menu/Goto view. Right-clicking in the HexView window generates the menu:



The top 4 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).

<u>'Goto view'</u>

The 'Goto view' takes the focus to the corresponding area in the view selected (see *Synchronized views/navigating the views, page 7- 6* for more information).

<u>'Synchronize views'</u>

When the 'Synchronize views' icon  is pushed in, then all open windows automatically follow the selection made.

See *Synchronized views/navigating the views, page 7- 6* for more information on 'Goto view' and 'Synchronize views'.

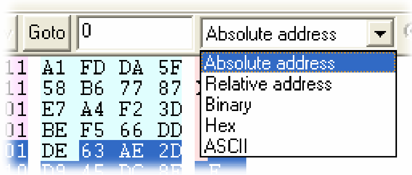
Highlighting a section. A section of data can be highlighted in any of the (visible) binary, hex or ASCII sections by clicking and dragging the mouse over the area.

The addresses of the selected area and the number of bytes selected are displayed in the status line at the bottom of the MTS4EA window. Whichever section is highlighted, the corresponding areas of the other sections are similarly highlighted.

Find absolute address. A specific address can be found in the video stream, either:

- an absolute address, from the start of the video file;
- a relative address, from the first byte of the currently-selected area - see *View stream hex…*  *Ctrl+H, page 7- 142*.

- 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 there is no address highlighted, the address found is the offset from 0.

The address to find is entered in the box next to the 'Next' button. This address can be in hex or decimal. If an invalid character is entered for an address (e.g. entering anything other than '0-9' or 'a-f' for a hexadecimal address) then the Find button is grayed out.

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

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

The 'Hex'/'Dec' radio button selects the format of the data being searched.

Find Binary/Hex/ASCII. These 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 irrespective of byte location; the Hex and ASCII searches are byte aligned.*

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

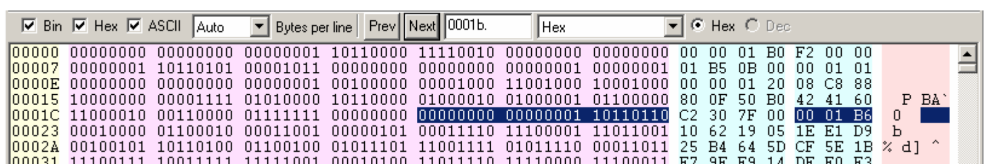
HINT. *The 'Shift+F3' key can also be pressed, to 'find previous'.*

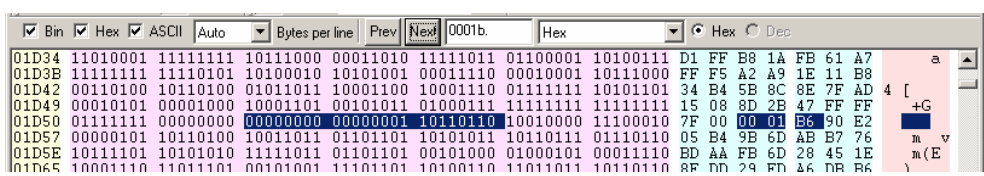
'Wildcard' searching. The period character ('.')can be used as a wildcard in searches.

The wildcard matches a single digit in the 'base' selected; i.e. the wildcard matches:

- one bit in the binary find
- a hex digit in the hex find
- an ASCII character in the ASCII find.

For example, entering '0001B.' as a hex string to find (that is, any character after the '0001B' pattern will find the following patterns in the selected example stream:





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'.

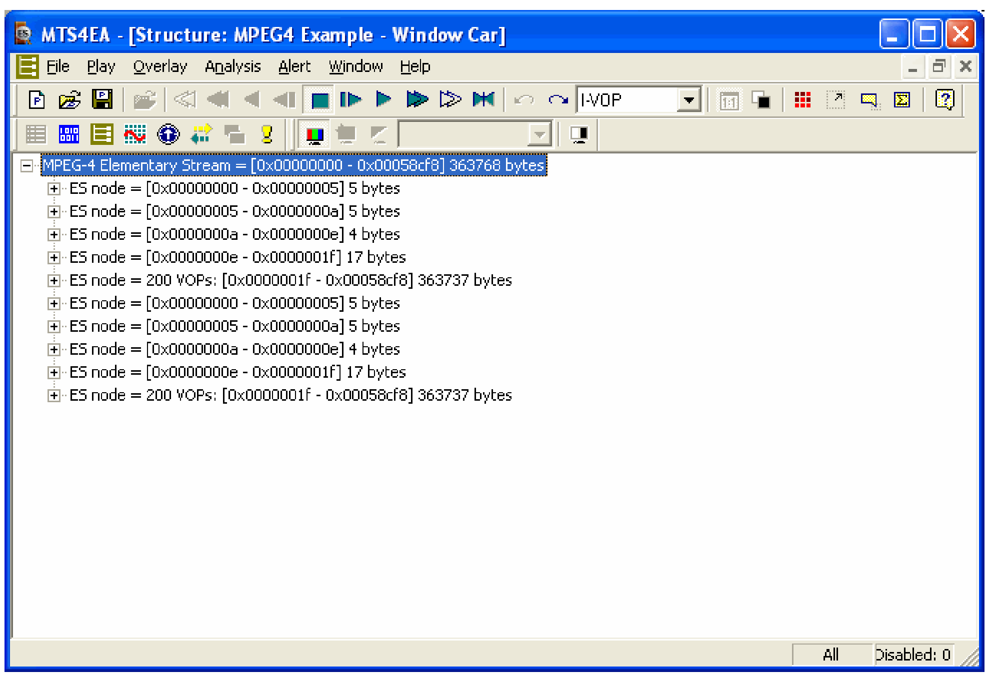
Conversion of hex<->decimal. If a hex value is entered into the find box:

then the 'Dec' radio button is clicked then this value is converted to hex (and vice-versa):

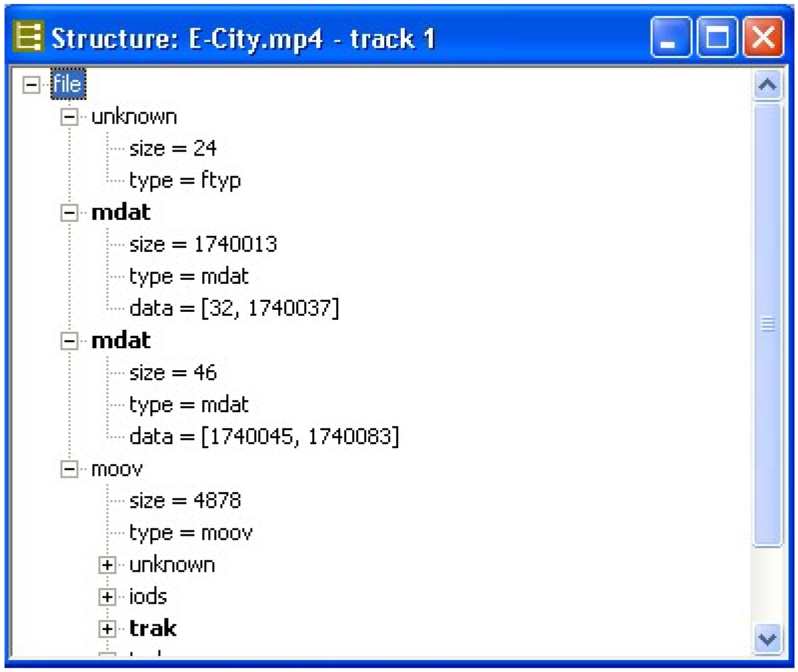
View file structure… Ctrl+R

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:



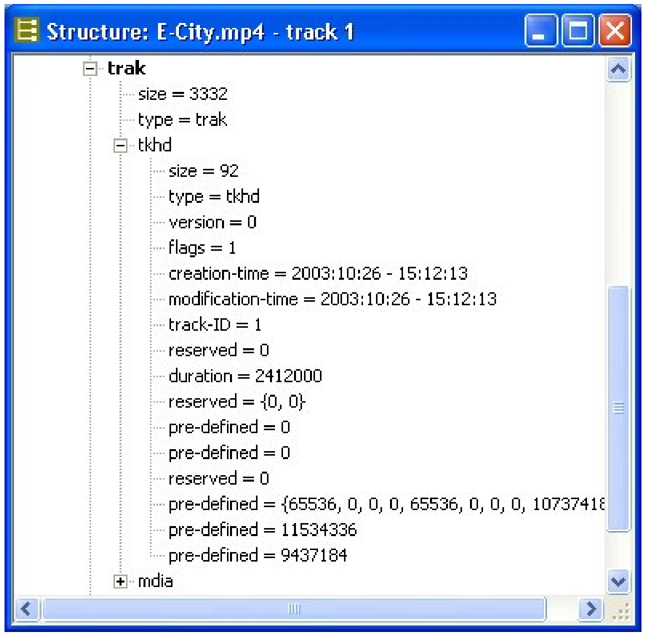
- 'container' files, e.g. MP4 and 3GPP files:



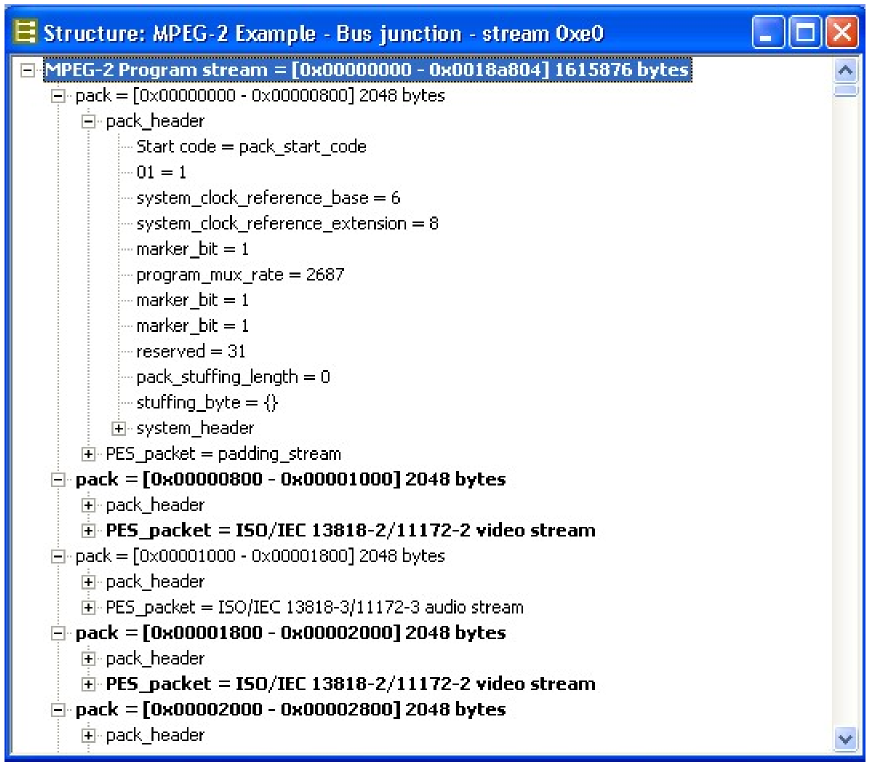
As shown above, when this window opens:

- the first two levels of atoms are expanded
- the video-related atoms are indicated in **bold** text.

Each branch of the structure can be clicked to expand it and see the atoms below (the '**trak**' atom has been expanded below):



Similar actions can be performed with MPEG-2 Program Streams, where the PES elements can be expanded/contracted:



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**);

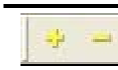
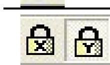
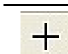
For **MPEG-2** the analysis is of:

- 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 *View buffer analysis..., page 7-151* (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 together in *View buffer analysis..., page 7-151*;
- the H.264/AVC HRD analysis is distinct and is explained in *View buffer analysis..., page 7-151*.

Buffer analysis toolbar icons.

Icon	Function
	Zoom in (+) and zoom out (-) centered on the center of the window - affected by the 'Lock X/Y' icons (see below)
	Fit all data into window
	Locate origin (zero), i.e. start of video 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, i.e. fill the analysis data in real-time, as the video is being decoded and scroll the window to the right
	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 - see the screen shot below
	Scroll/pan (the cursor changes to show the scroll/pan direction) - this is affected by the 'Lock X-Y' icon - see above
	Zoom in/zoom out centered on the location of this cursor - this is affected by the 'Lock X-Y' icon - see above. Press the <shift> key to zoom out

Right-click pop-up menu/Goto view.

	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		Goto the origin (time = 0)
X-Axis lock		Lock the X-axis when zooming/scrolling/pan
Y-Axis lock		Lock the Y-axis when zooming/scrolling/pan
Autoscroll		Autoscroll the window to follow the video frames being decoded
Move tool		Move the window left/right/up/down
Measure tool		Measure the values at the center of the '+'
Zoom tool		Zoom in/out - centered at the cursor
Goto view	}	see following paragraphs
Synchronise views	}	see following paragraphs

The functions listed above from 'Zoom-in' to 'Zoom tool' inclusive are all explained above in *View buffer analysis..., page 7-151*.

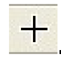
<u>'Goto view'</u>

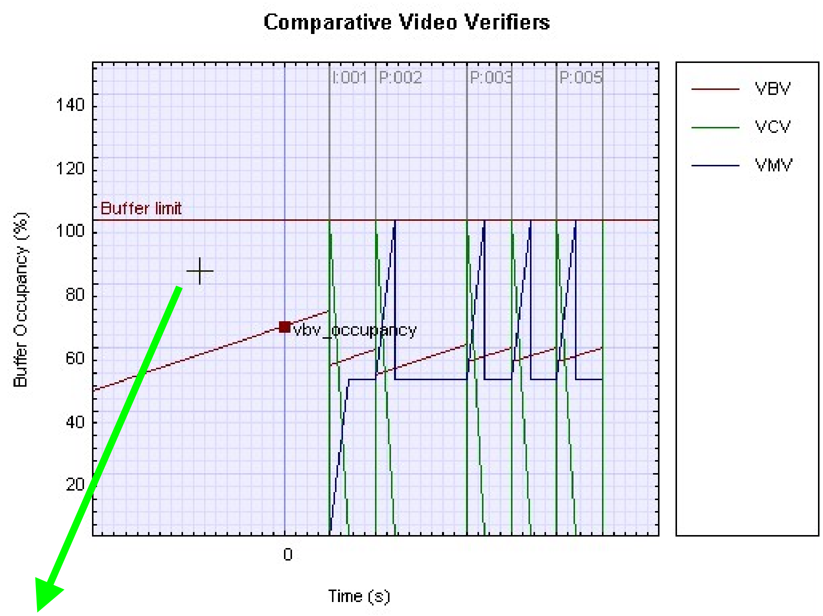
The 'Goto view' takes the focus to the corresponding area in the view selected (see Synchronized views/navigating the views, page 7-6 for more information).

<u>'Synchronize views'</u>

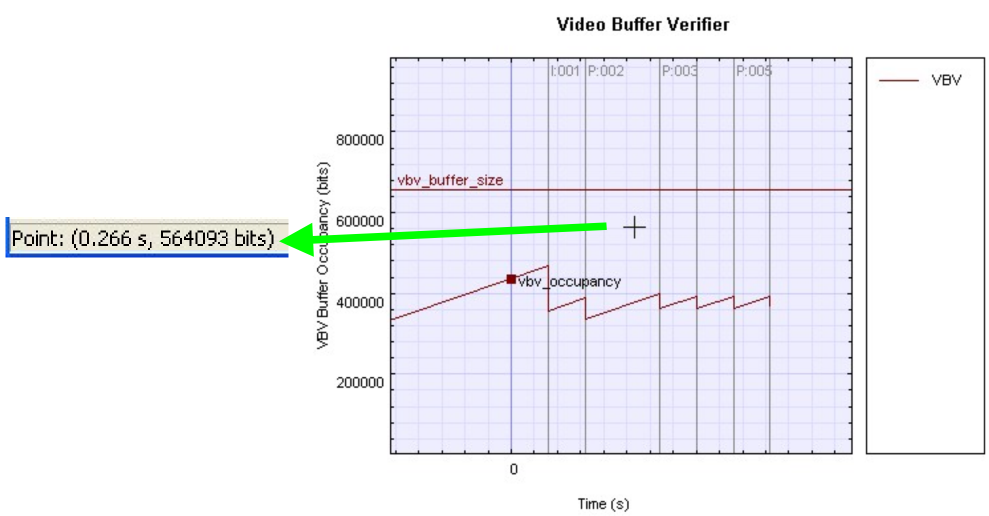
When the 'Synchronize views' icon is pushed in, all open windows automatically follow the selection made.

See Synchronized views/navigating the views, page 7-6 for more information on 'Goto view' and 'Synchronize views'.

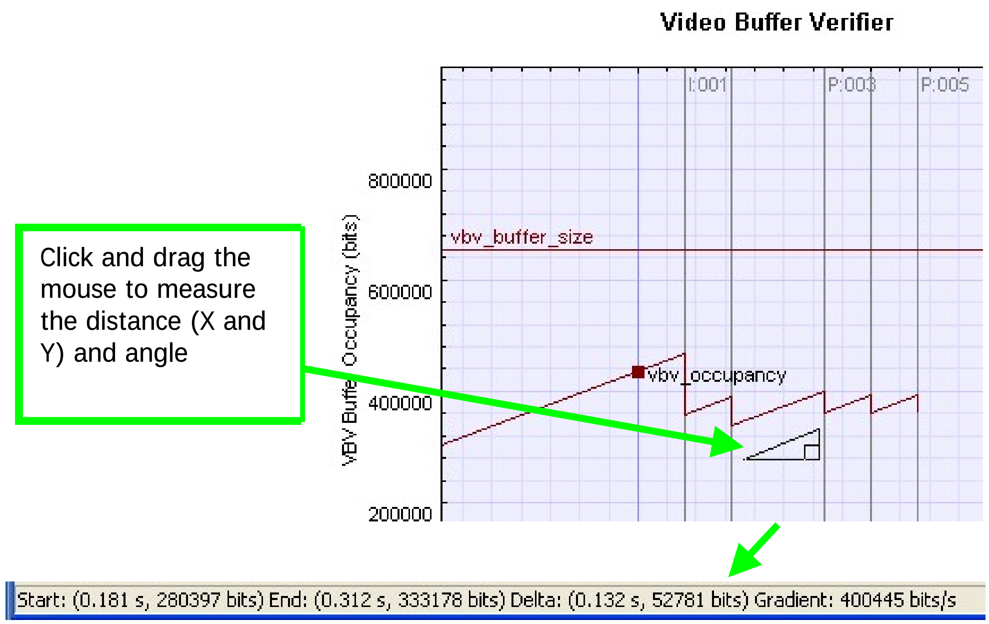
Using the measure tool . Moving the measure tool over the graph displays the values in the status line:



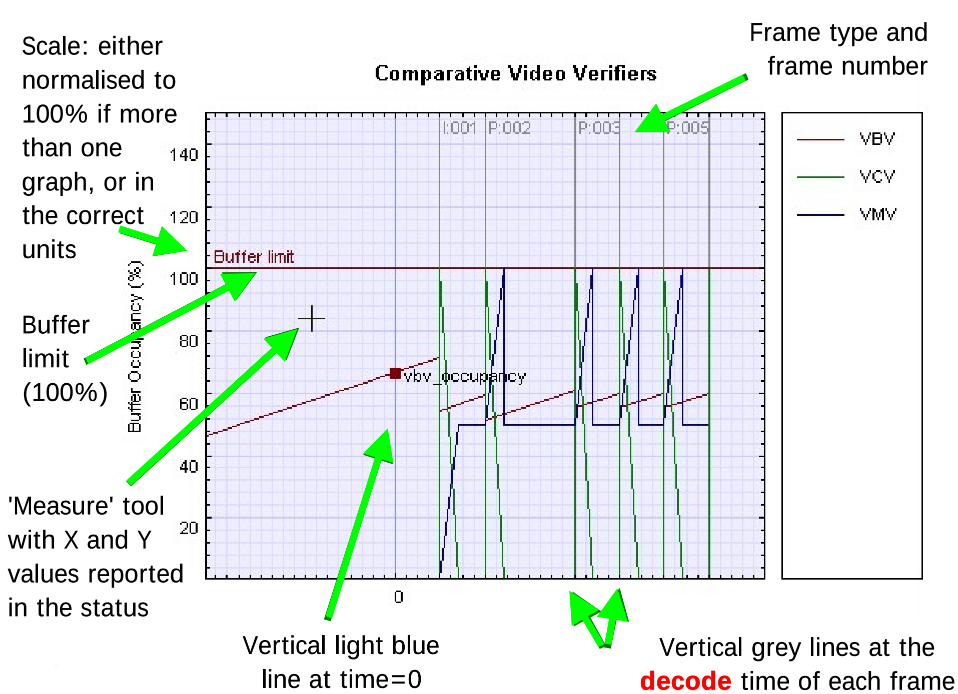
The left-hand value is the X-axis (time) value in seconds, to the nearest millisecond; the right-hand is the Y-axis value:
- either in percentage value, if there is more than one graph
- or in the correct units, if there is only one graph (as below)



(Only the VBV is shown here)

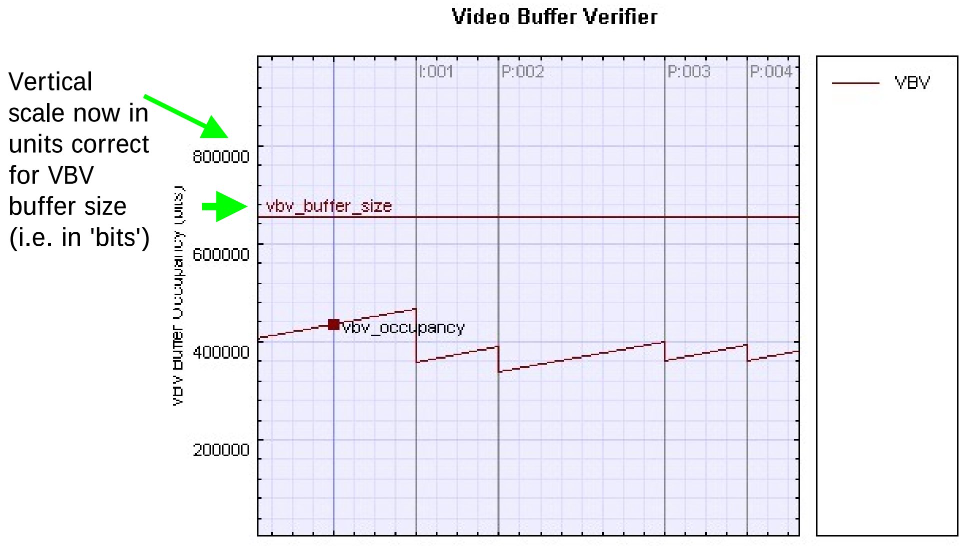


Buffer analysis graph area.



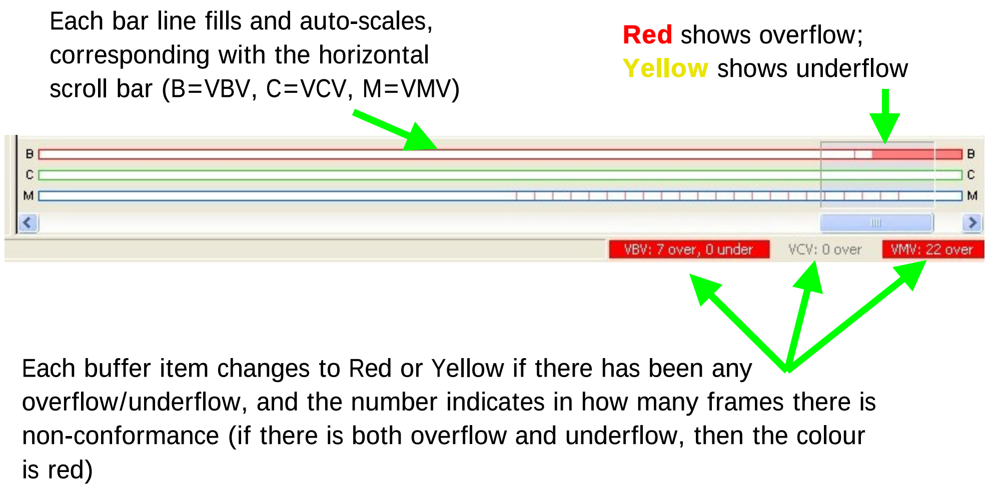
The left axis displays:

- values normalized to 100%, if there is more than one graph, or
- values appropriate for that graph - see below

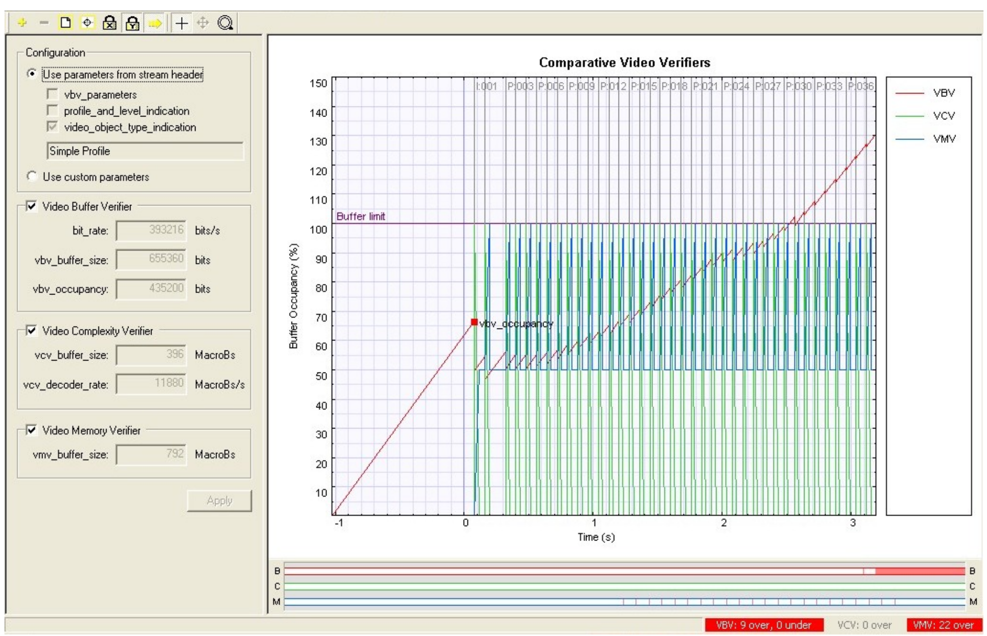


When the buffer data is too wide/too high for the current window, then scroll bars appear at the bottom/right (as appropriate).

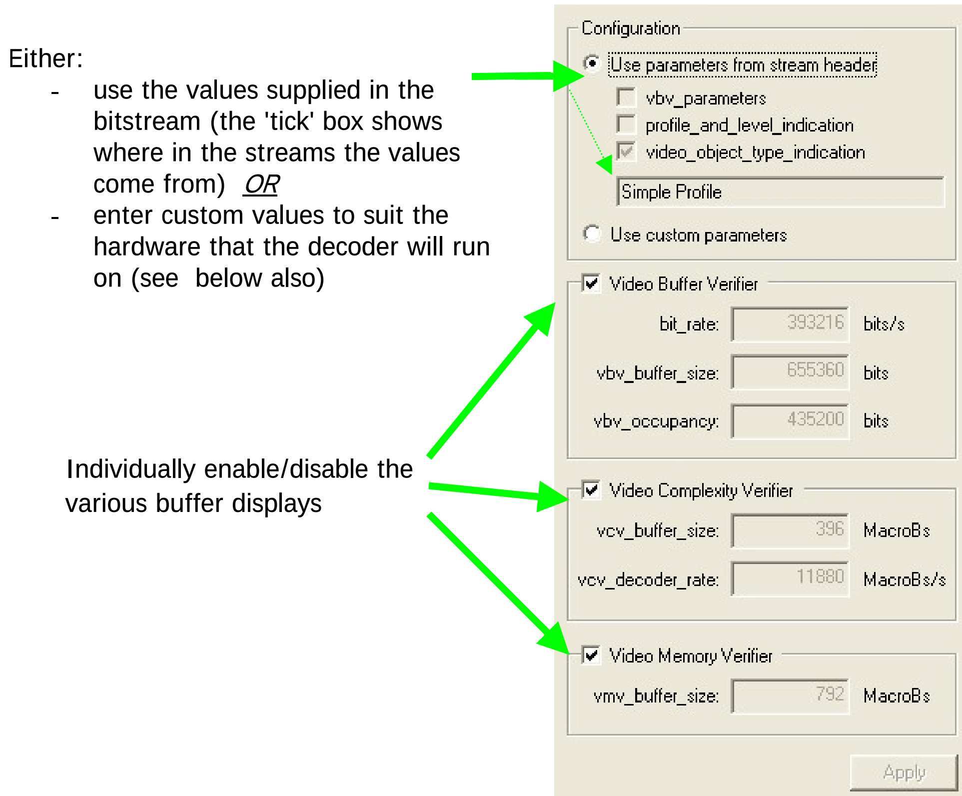
Buffer analysis alerts/scroll bar area. The buffer overflow/underflow is indicated in the Graph window as shown below:



MPEG-4 and MPEG-2 Buffer analysis. All the example screen-shots 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.

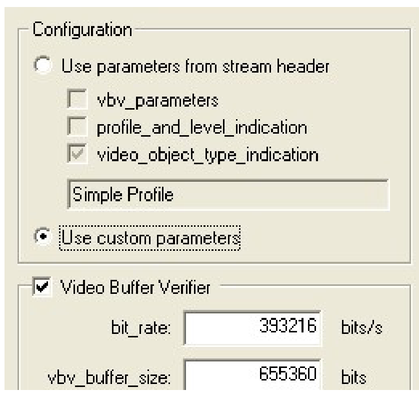
The boxes below the title 'Use parameters from stream header' show where the values used have come from: in the example above, there were no values specified in the '`vbv_parameters`' and nor was there a '`profile_and_level_indication`' so the maximum value implied by Simple Profile as given in the '`video_object_type_indication`' 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`'*

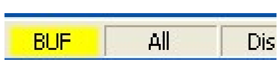
Use custom buffer parameters: MPEG-4 and MPEG-2. Custom parameters can be entered for a variety of reasons, for example:

- 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).*

NOTE. *The values entered as Custom parameters affect both the alert pop-up warnings and the warnings that are reported in the Trace files, i.e. the Custom parameters are the values used to trigger these warnings.*

HINT. *To reset the values entered to the values specified by the bitstream, simply reselect the radio button 'Use parameters from 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 (i.e. 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).*

It is planned that DPB analysis will be added in a future version of MTS4EA.

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 in *View buffer analysis..., page 7-151*; 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 meaning of the column headings are:

<u>Index</u>

The bucket or index number

<u>CPB size (bits)</u>

('CPB' is Coded Picture Buffer)

The size in bits of the Coded Picture Buffer.

<u>Bit rate (bits/s)</u>

The bit rate of the hypothetical transmission channel

<u>Initial delay (s)</u>

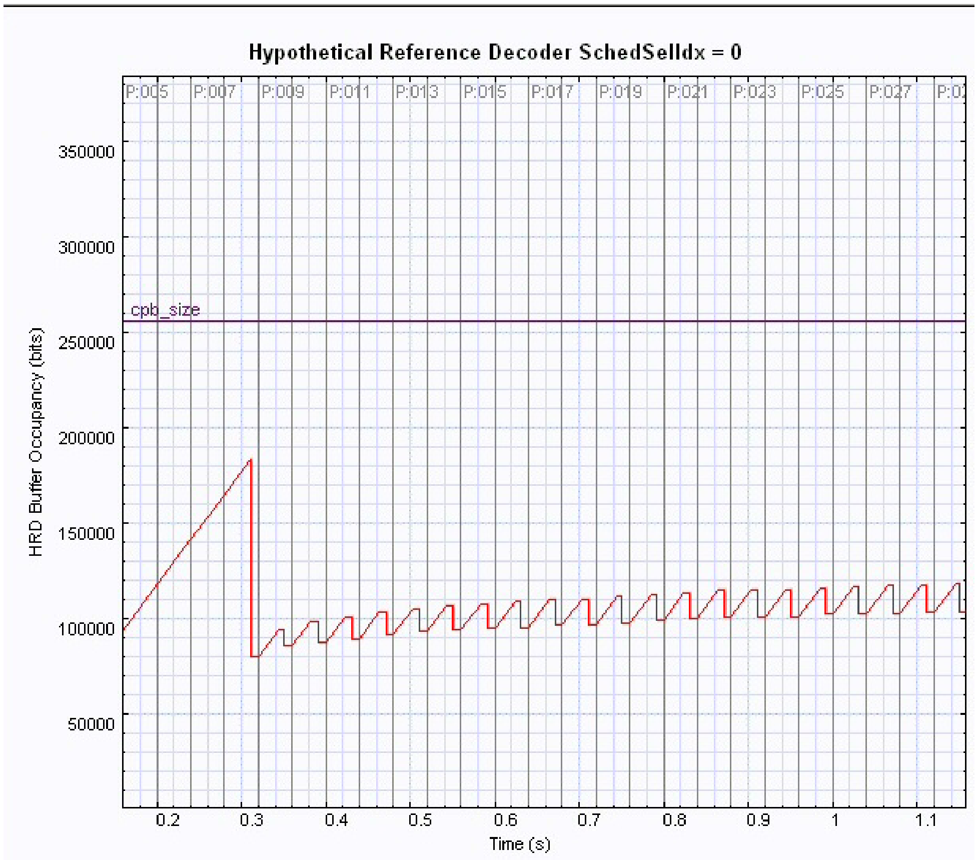
The initial delay of the hypothetical transmission channel before the first frame is decoded

<u>CBR flag</u>

('CBR' is Constant Bit Rate)

The status of the Constant Bit Rate 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):



'cpb_size' is the Coded Picture Buffer size.

The areas of the graph display are explained in *View buffer analysis..., page 7-151*. 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 frames.

HRD buffer overflow/underflow indication: H.264/AVC. HRD buffer analysis overflow and underflow is indicated:

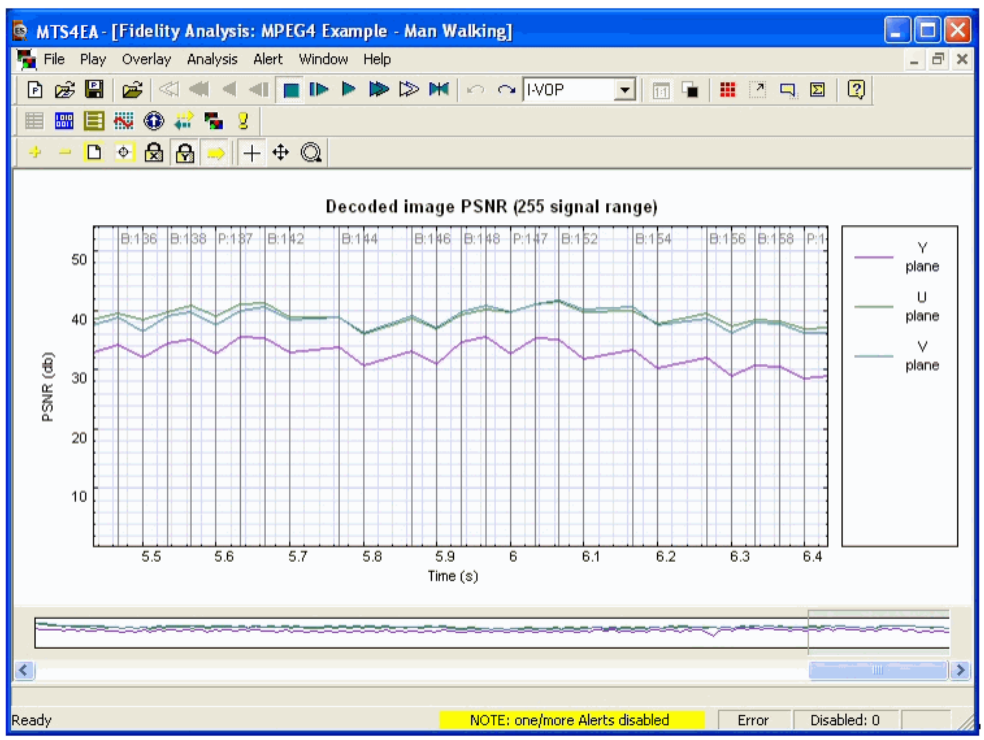
- in the bar line below the graph area
- in the status bar and
- in pop-up alerts.

The bar line and status bar indications are the same as overflow and underflow indications for MPEG-4/MPEG-2 (see *View buffer analysis..., page 7-151*).

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, page 7-167*).

View fidelity analysis...

This displays the fidelity analysis graph view:



The fidelity analysis graph view is similar to the graph view as used for buffer analysis in MPEG-4, MPEG-2 and H.264/AVC (see also *View buffer analysis..., page 7-151)*:

- it has the same control icons
- the graph area display is similar, with the same method display of frame types, frame numbers, frame times
- with the same means of measuring angles
- and the same right-click pop-up menu

The fidelity analysis results are shown separately for Y, U and V planes:

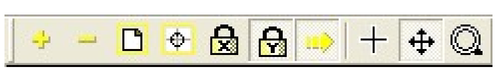
- Y-plane in **Mauve**,
- U-plane in **Green**,
- V-plane in **Blue**.

HINT. *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.*

HINT. *The first 9 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.*

NOTE. *Where the frame rate is incorrectly set for the YUV reference file, then the correspondence between the encoded video frames and the YUV source frames is lost: this will substantially reduce the fidelity analysis values.*

Fidelity analysis view icons toolbar.



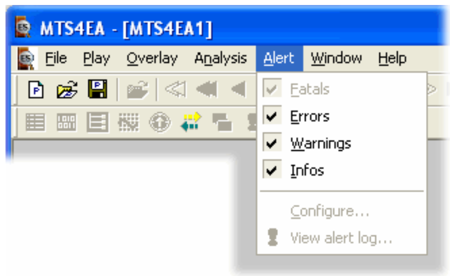
The icons have the same functions as the icons on the buffer analysis toolbar: see *View buffer analysis..., page 7-151*.

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 in *Fidelity enable..., page 7-132*.

Alerts 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 *Explanation of Fatal/Error/Warning/Info display, page 7-167* regarding the step-back buffer cache). However, the alerts found will be highlighted in the Alert log view (and if this view is not open, a warning is given that there are Alerts occurring) - see *Explanation of Fatal/Error/Warning/Info display, page 7-167*.

NOTE. *For some Alerts, extra information is provided about the section of the video which has generated the error, and the standard concerned if it is standard-specific. See General codes used in Trace Files and Alerts, page 8-1 for detailed information*

Alert levels

There are 4 different levels of Alerts: ‘Fatal’, ‘Error’, ‘Warning’ and 'Info' (in order of decreasing severity). See *Explanation of Fatal/Error/Warning/Info display, page 7-167* for a description of each of these.

NOTE. *The ‘Fatal’ alert is always set (and cannot be switched off).*

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

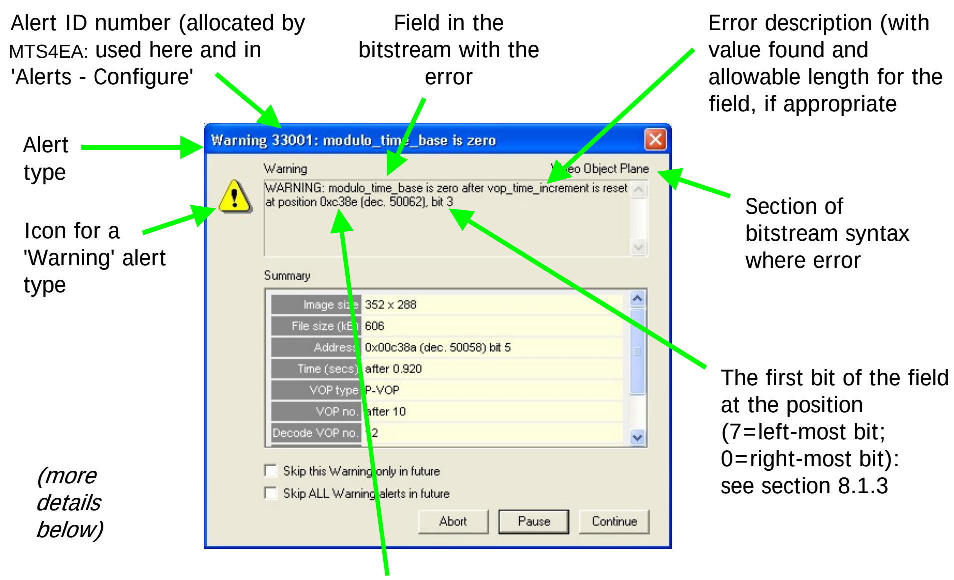
When a less severe Alert is set, the more severe Alerts are set automatically. (For example, setting 'Warnings' on automatically sets 'Errors' and 'Fatal' **on**.)

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 *Explanation of Fatal/Error/Warning/Info display, page 7-167*.

Explanation of Fatal/Error/Warning/Info display

Information at top of alert pop-up.



Byte position in the bitstream where the error was found: 0x…is the hexadecimal address, '(dec...)' the address in decimal.

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 Explanation of Fatal/Error/Warning/Info display, page 7-167 for more information.*

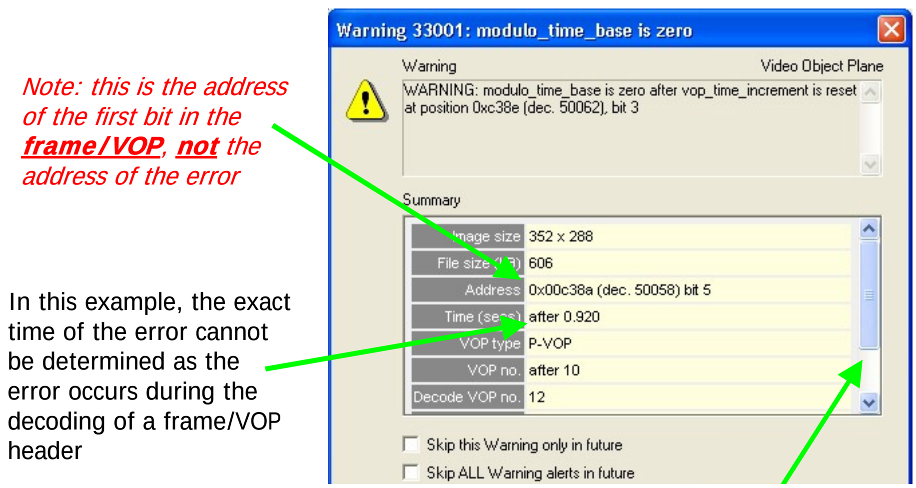
NOTE. *For all Fatal, Error, Warning and Info displays:*
- the 'bitstream position' of the error is the number of bytes from the beginning of the video bitstream, where the first byte is byte zero;
- the position is displayed in hexadecimal (0x...) and decimal (dec. ...);
- the 'bit start' is the first 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.

NOTE. *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 General codes used in Trace Files and Alerts, page 8-1 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, page 7-57* for a detailed explanation of the fields and data provided.



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, i.e. 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

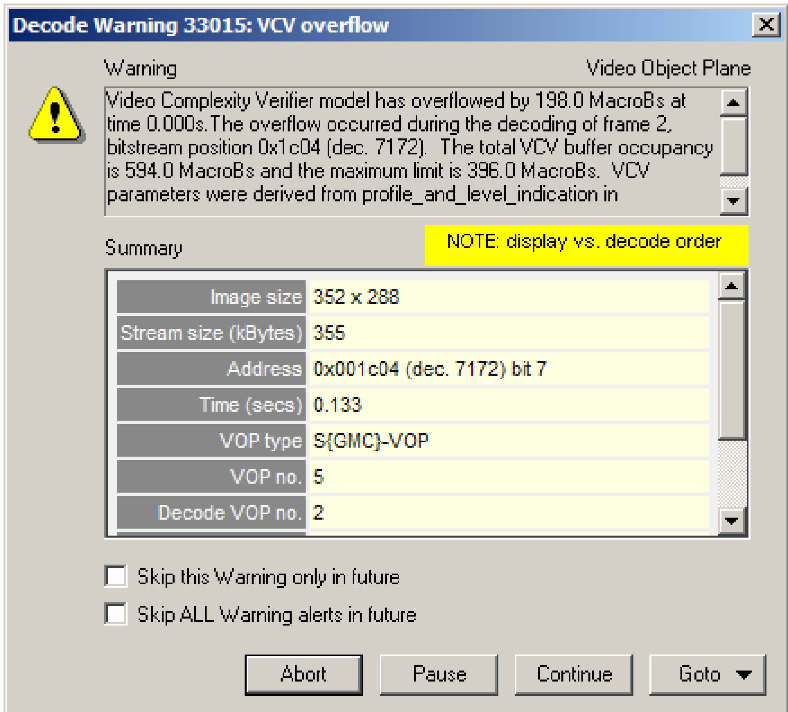
- order of data in **displayed**:

Display frame number	-	1	2	3	4	5
Frame type	-	I	B	B	P	P
Errors 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 _not_ a B-frame/B-VOP.

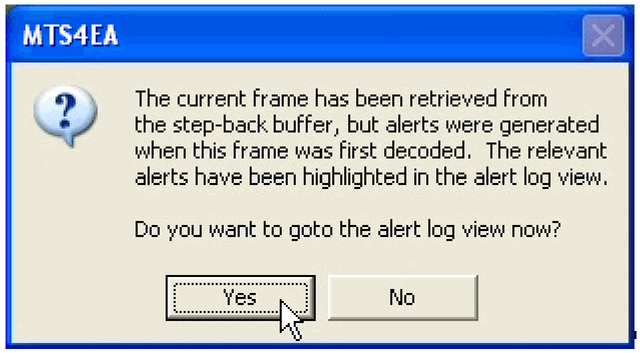
NOTE. *Note that the yellow warning can also appear in other circumstances, where the alert has occurred some time after a displayed frame.*

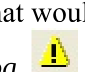
Pop-up alerts when seeking forwards/backwards through data in the step-back buffer (cached). When the video is decoded, pop-up alerts may occur. The results of the decoding are stored in a cache, the 'Step-back buffer' - see *Decoder options..., page 7-44* for more information.

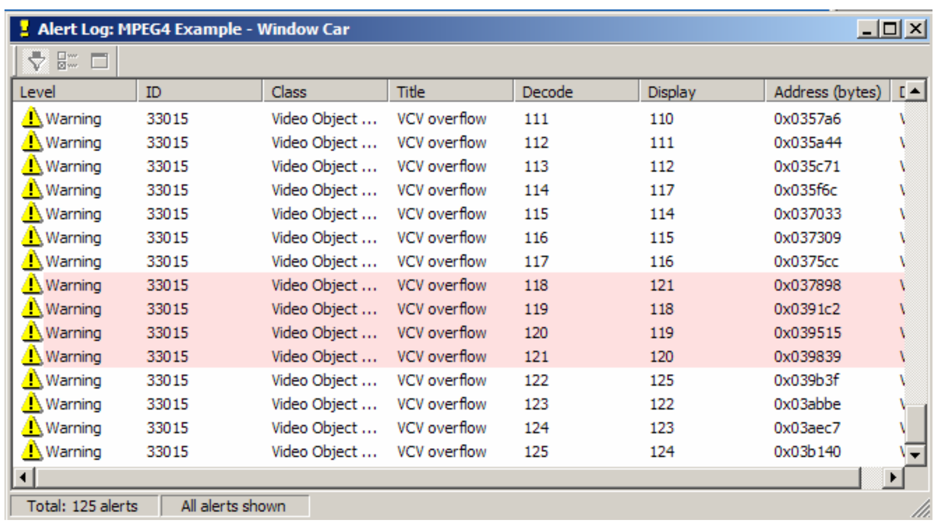
It is possible to step forwards/backwards over the frames where these alerts occur. To avoid a continual repetition of the pop-ups (and potentially many such alerts appearing), the pop-up alerts are **not** displayed when seeking forwards/backwards in these situations.

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

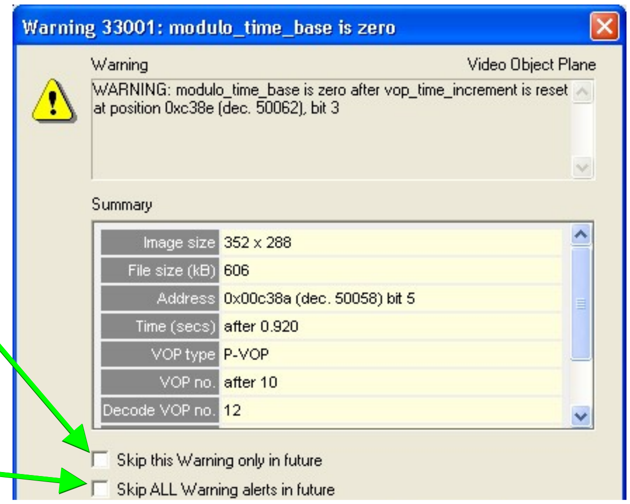
- either, there is a pop-up message, at most once per seek forwards or backwards:



- or, 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* , *page 7-178* for more information).



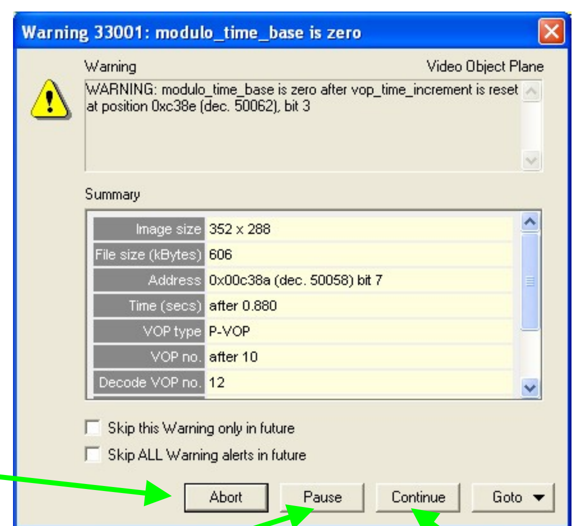
Check box options.



Tick/check this box to not get an alert for this *specific* Warning only (but other 'Warnings', 'Errors' and 'Fatals' still trigger an alert)

Tick/check this box to not get an alert for *any* 'Warning' level alerts ('Errors' and 'Fatals' still trigger an alert)

Abort, Pause, Continue.



'Abort' stops the decoding immediately, at the end of the current alert, and enter 'stop' mode

'Pause' keeps decoding to the end of the current frame/VOP and then enter 'stop' mode

'Continue' decoding normally (and keep playing, in normal, fast or step modes)

HINT. *'Abort' can also be done by pressing the 'Esc' (escape) key.*

Goto button. This takes the focus to the position in the selected view which corresponds most closely to the location of the error. See *Synchronized views/navigating the views, page 7-6* 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 a particular MacroBlock	the MacroBlock containing the error (highlighted with a yellow/black dotted square)
	If not (i.e. the error is in a frame header or file header)	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	
Navigator	The frame containing the error or the frame which immediately follows the error (if the error is in a header)	
Buffer	The frame containing the error or the frame which 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	
Structure	The syntax element corresponding to the location where the error has been found	

Description of Alert levels

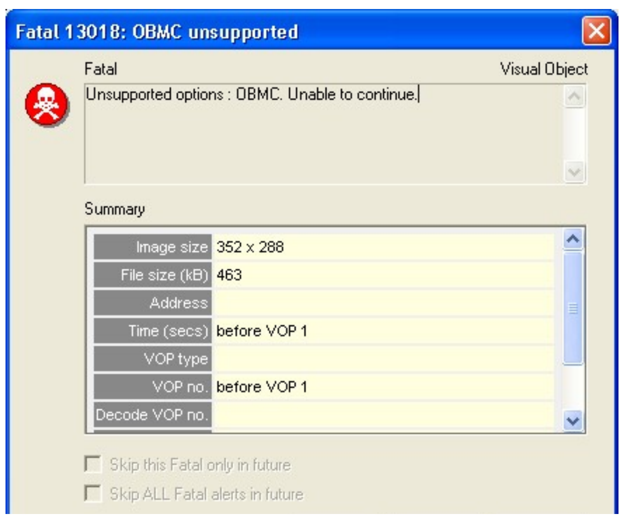
Fatal. A Fatal error occurs when there is something encountered in the bitstream which cannot be decoded by MTS4EA.

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:

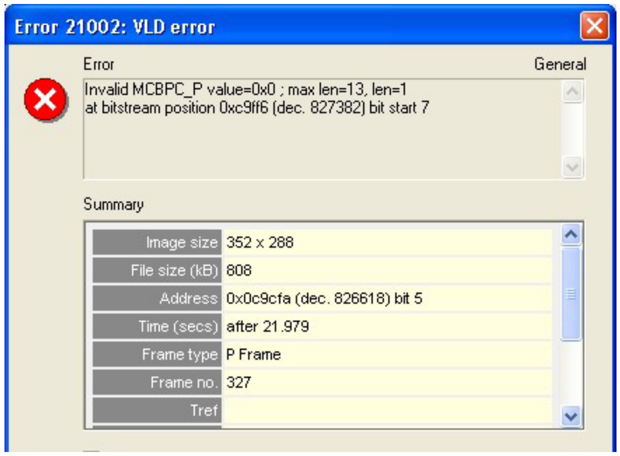
- severe errors in the syntax of the bitstream, which make it completely unintelligible to MTS4EA , or
- options being selected in the video bitstream that are not yet supported by MTS4EA.

For example, the Fatal error below was caused by the OBMC flag in an MPEG-4 bitstream being set (on), i.e. 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.)



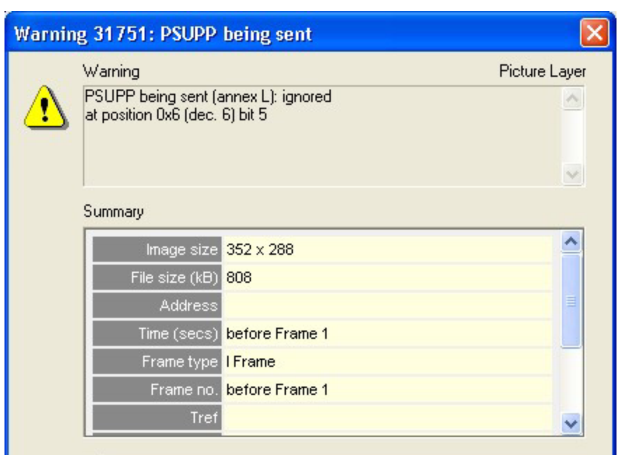
Errors. When this item is selected/checked, the decoder will stop at any point where there is an Error in the bitstream.

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):



Warnings. When selected/checked, this item will provide a number of warnings which 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)



Infos. 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, or
- 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 over-rides the enabling of specific alerts but NOT the disabling of specific alerts.*

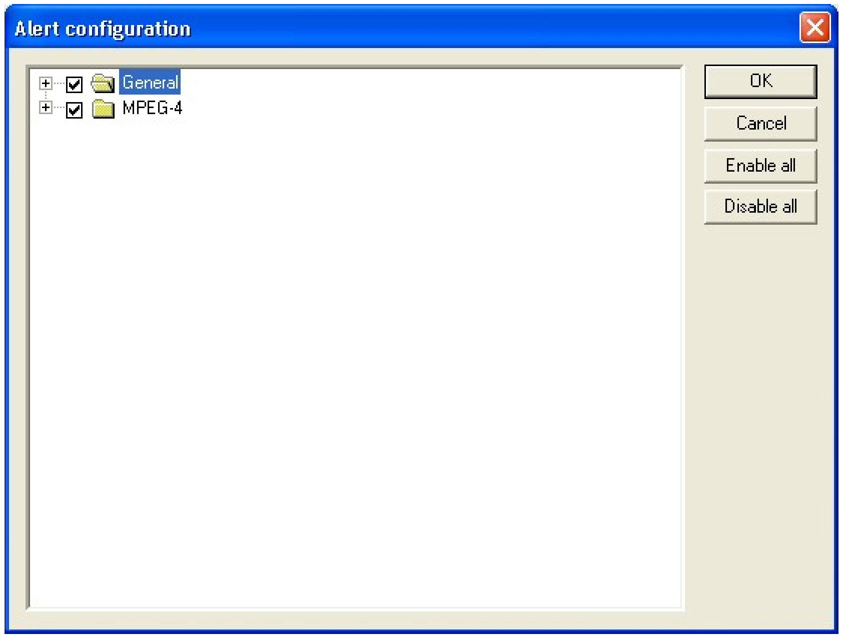
This means, for example:

- that a specific 'Warning' alert can be enabled/disabled, but ***not*** 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, page 7-185*.
- 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 in *Status bar indication of Alert status, page 7-185*.

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 is planned for future versions of MTS4EA.*

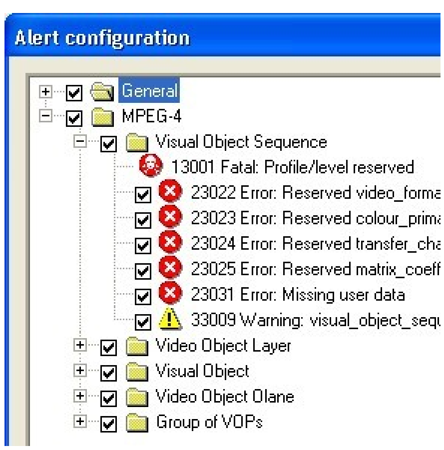
Enable/disable specific alerts.



This displays two folders at the top level:

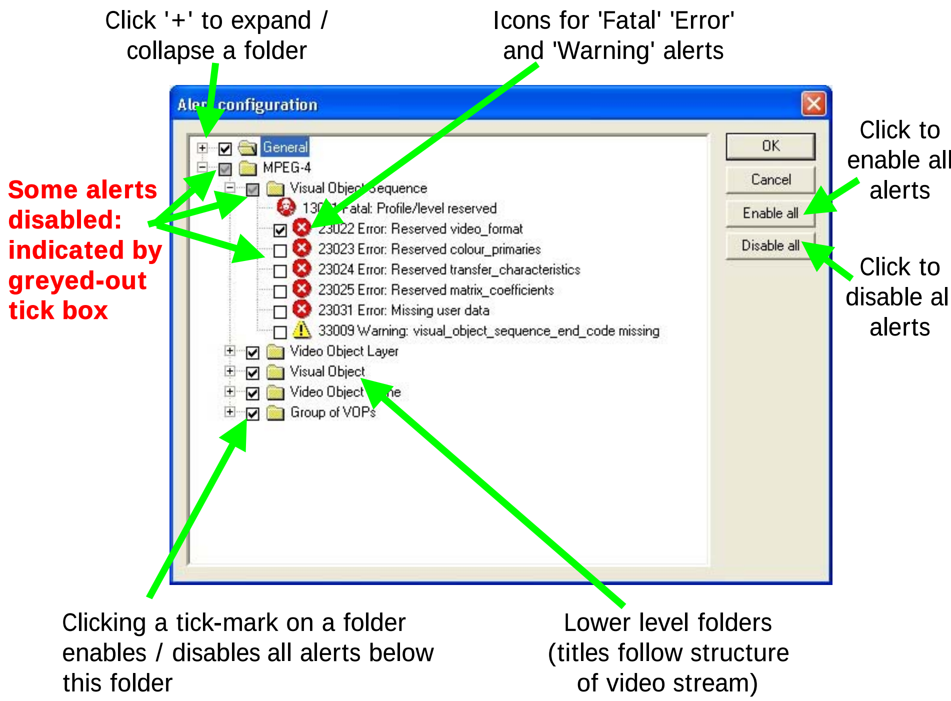
- 'General' alerts, which can apply irrespective of the standard concerned
- a 'standard folder', i.e. 'MPEG-4', 'H.263' or 'H.261' which contains alerts specific to the standard concerned

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



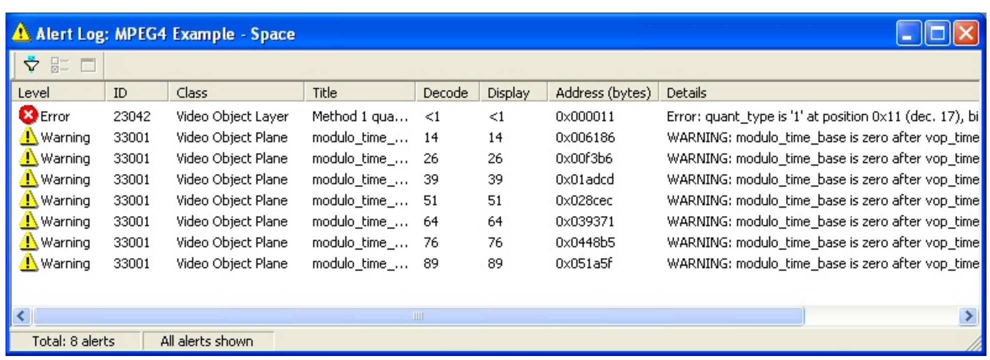
Elements of the Alert configuration window.

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 and column titles.

Alert filter icon

See *Alert log* , *page 7-178*.

Configure alerts icon

Clicking on this icon opens the Alert configuration dialog box - see *Configure Alerts* , *page 7-176*.

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 *Alert log* , *page 7-178*.

If this icon is grayed out, it can be enabled by selecting any alert that is shown.

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 where this error has been found.

See *Explanation of Fatal/Error/Warning/Info display, page 7-167* for more information on decoded versus displayed frame numbers, in bitstreams which include B-frames/B-VOPs.

NOTE. *If '<1' is displayed, this means that the error has occurred before the decode of the first frame, i.e. the error has occurred in the header.*

Display

The number of the displayed frame where this error is shown.

See *Explanation of Fatal/Error/Warning/Info display, page 7-167* for more information on decoded versus displayed frame numbers, in bitstreams which include B-frames/B-VOPs.

NOTE. *If '<1' is displayed, this means that the error has occurred before the decode of the first frame, i.e. 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 which have been extracted from container files (e.g. MP4 files, 3GPP files, MPEG-2 Packet Streams) then this address is the byte address in the extracted video stream, not in the overall container file.*

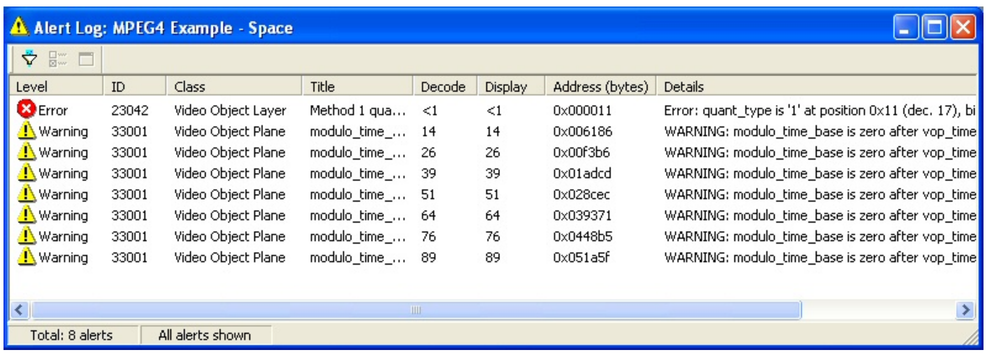
Details

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

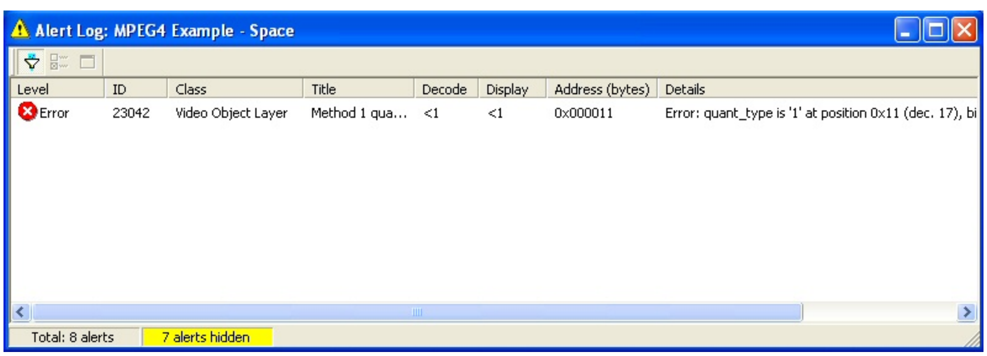
Showing/hiding alerts (alert filter). In some cases there can be hundreds or thousands of alerts. It can be that these are 'known issues' and therefore the user does not want to list these in the Alert log window (as these would clutter the window unnecessarily).

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* , *page 7-176* 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, 7 alerts are hidden out of the total of 8 alerts)

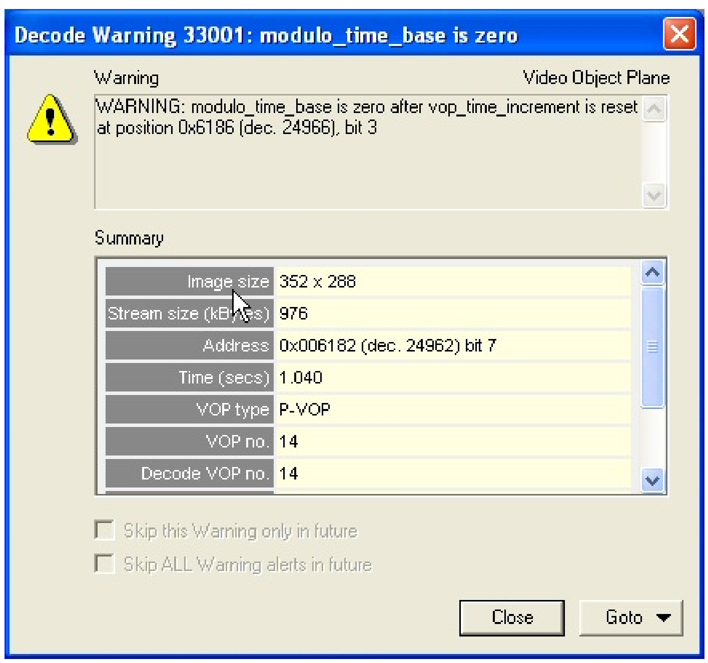
The Alert log filter can also be switched on/off by right-clicking and selecting the option from the pop-up menu.

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 (that is, 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 .

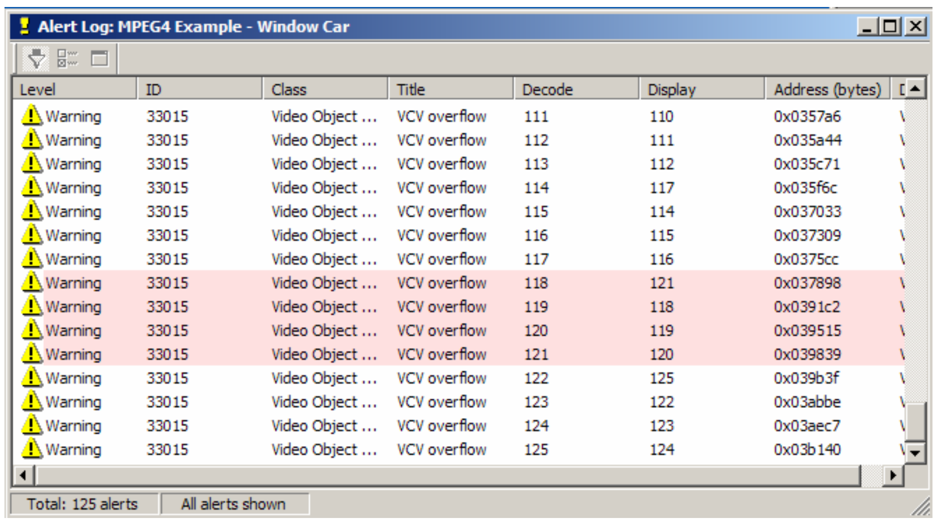
When this is done, a display is shown very similar to the original pop-up alert display:



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 *Decoder options..., page 7-44* for more information.

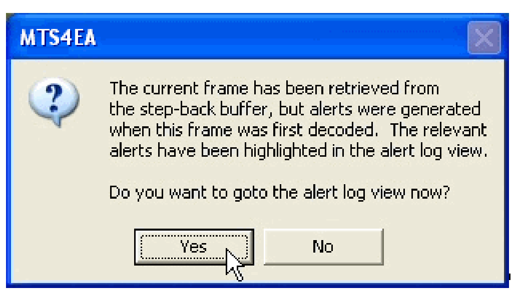
When stepping backwards/forwards through video which is held in the step-back buffer, to avoid continually having known alerts appearing (which then require to be cleared or skipped) MTS4EA simply highlights the alerts relevant to the area being skipped in the Alert log. For example, if skipping forward 100 frames and there would normally 5 pop-up alerts during the decoding of the 100 frames, then MTS4EA will highlight the 5 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:



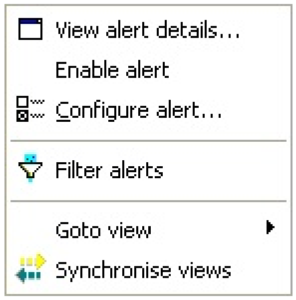
Level	ID	Class	Title	Decode	Display	Address (bytes)
Warning	33015	Video Object ...	VCV overflow	111	110	0x0357a6
Warning	33015	Video Object ...	VCV overflow	112	111	0x035a44
Warning	33015	Video Object ...	VCV overflow	113	112	0x035c71
Warning	33015	Video Object ...	VCV overflow	114	117	0x035f6c
Warning	33015	Video Object ...	VCV overflow	115	114	0x037033
Warning	33015	Video Object ...	VCV overflow	116	115	0x037309
Warning	33015	Video Object ...	VCV overflow	117	116	0x0375cc
Warning	33015	Video Object ...	VCV overflow	118	121	0x037898
Warning	33015	Video Object ...	VCV overflow	119	118	0x0391c2
Warning	33015	Video Object ...	VCV overflow	120	119	0x039515
Warning	33015	Video Object ...	VCV overflow	121	120	0x039839
Warning	33015	Video Object ...	VCV overflow	122	125	0x039b3f
Warning	33015	Video Object ...	VCV overflow	123	122	0x03abbe
Warning	33015	Video Object ...	VCV overflow	124	123	0x03aec7
Warning	33015	Video Object ...	VCV overflow	125	124	0x03b140

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:



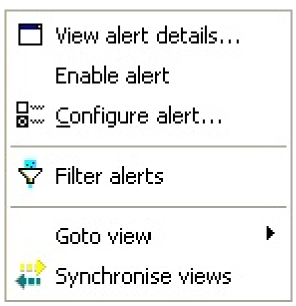
The functions of this menu are:

View alert details...

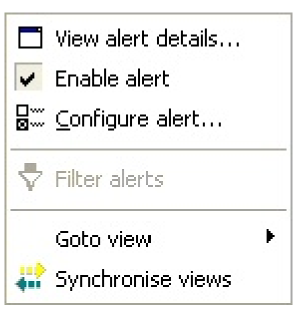
See *Alert log , page 7-178.*

Enable alert

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



Alert disabled



Alert enabled

HINT. *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*, *page 7-176*.

Filter alerts

See *Alert log*, *page 7-178*.

Goto view

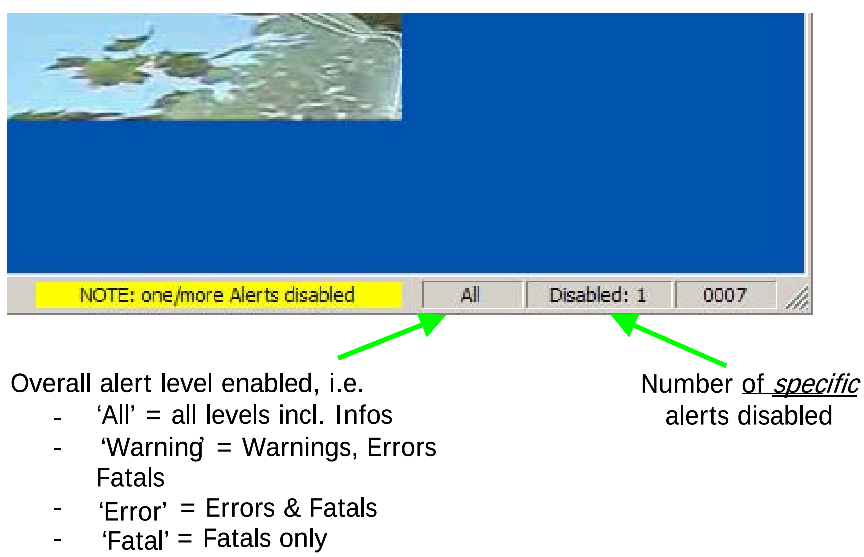
See Synchronized views/navigating the views, page 7-6.

Synchronize views

See Synchronized views/navigating the views, page 7-6.

Status bar indication of Alert status

The status bar displays the status of the alerts.



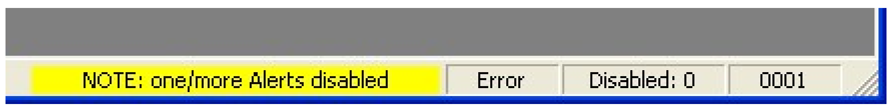
One or more alerts disabled. When one or more alerts are disabled, either because:

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

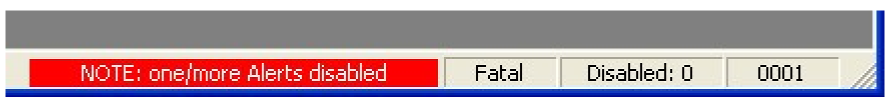
then the status bar changes as below.

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

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



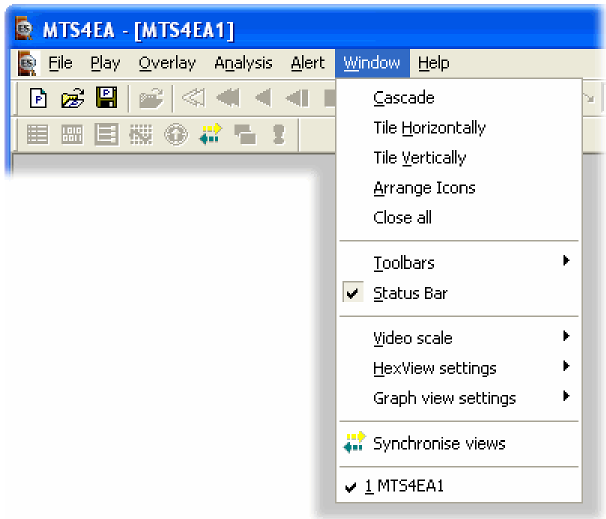
Overall alert level: 'Errors' disabled



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

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

Window Menu

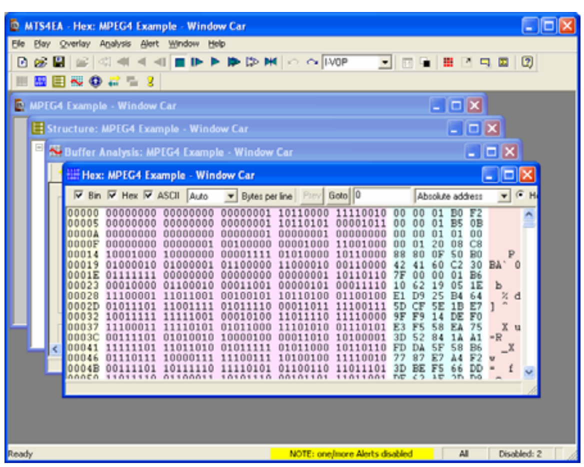


The commands on this menu are typical of a standard Windows application, such as arranging windows; list of open windows.

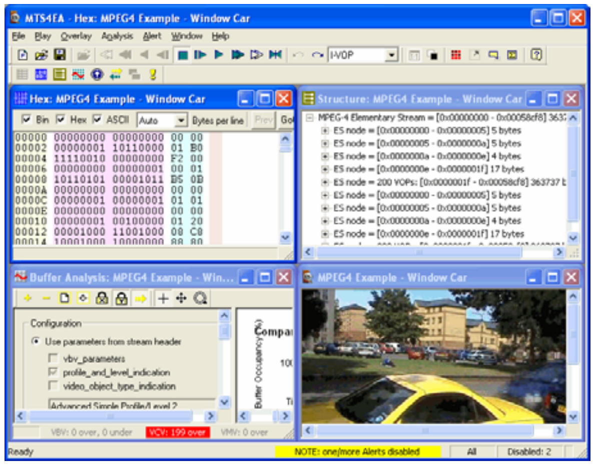
Cascade, Tile, Arrange Icons, Close All

These function as per standard Windows commands:

- 'Cascade' arranges the open windows in a 'cascade':



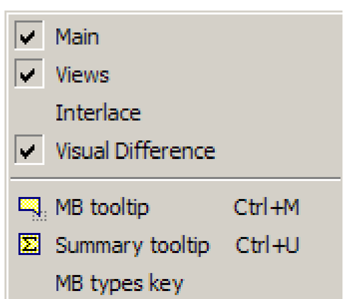
- 'Tile' arranges the open windows in tile fashion:



- '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 toolbar. 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 *Icon Toolbars, page 7-196* for more information.

Views toolbar. The Views toolbar is shown by default. See *Icon Toolbars, page 7-196* for more information.

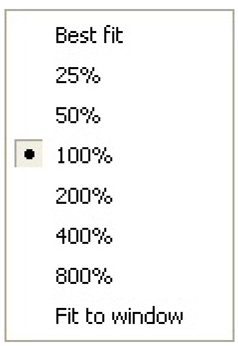
Interlace toolbar. The Interlace toolbar is not shown by default: it is always switched on when a stream which is interlaced (or could be interlaced - in H.264/AVC this may not be known until later in the stream) is opened. See *Context-sensitive Toolbars/Tooltips, page 7- 201* for more information.

Differences toolbar. The Differences toolbar is now shown by default: it is not shown until 'Visual differences' is enabled on the 'Overlay menu. See *Visual difference, page 7- 93* for more information.

Status Bar

When selected (which is the default setting), the Status Bar is displayed at the bottom of the window. When not selected, it is hidden. See *Status Bar, page 7-206* for more information.

Video scale

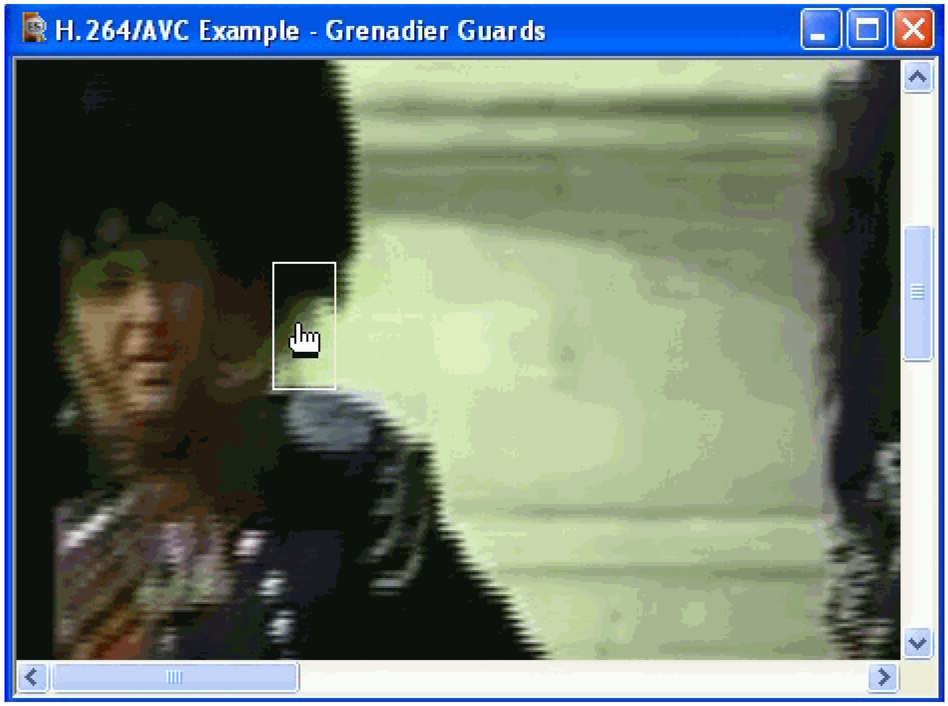


This sets the size that the decoded video appears in the video window.

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.

Percentages. 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:

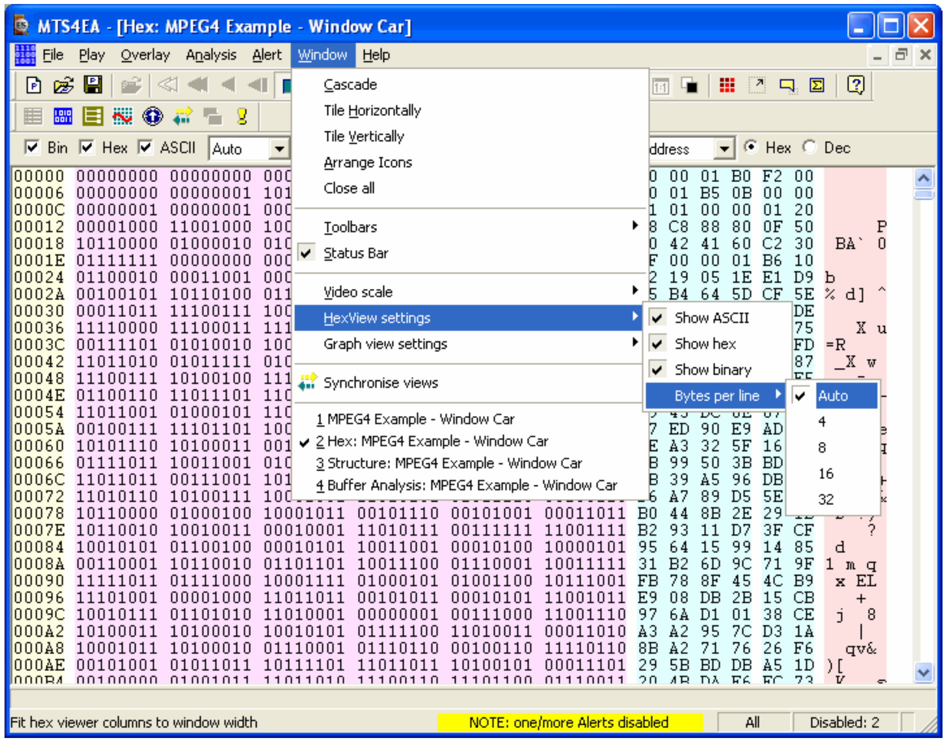


(In this example, zooming in to an interlaced bitstream: this is why Interlace artifacts are visible and the MacroBlock selection is two MacroBlocks high)

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 equalling 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 4 options:

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

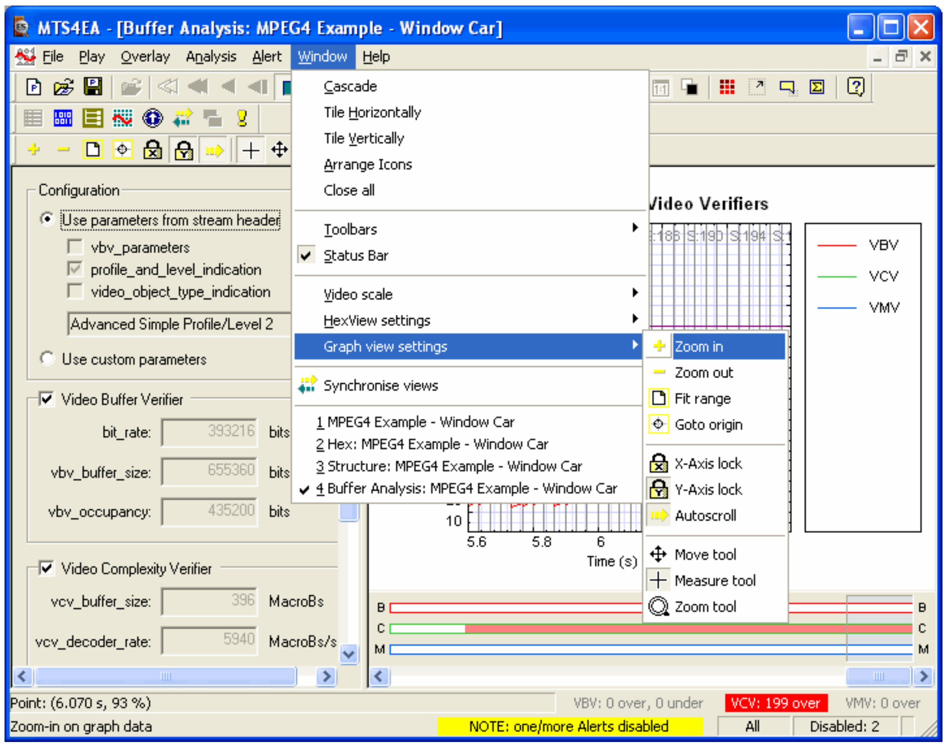
Show ASCII, Show hex, Show binary. These options individually switch on/off the display of the relevant area of the HexView.

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 are 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 10 options which control how the data appears on the Buffer Analysis graph (the term 'Graph view' has been used as further graphs will be added to MTS4EA, and this menu item will apply to them all):

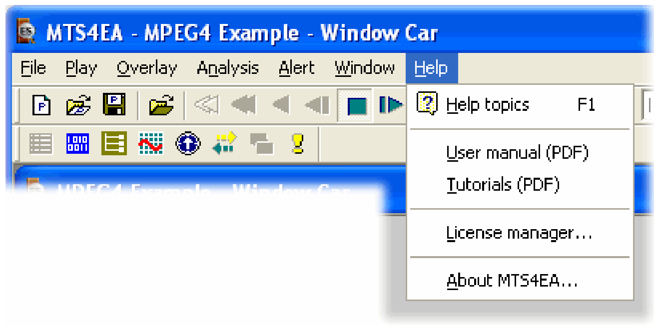
	Equivalent toolbar icon	Function
		Zoom in (centered on current window)
		Zoom out (centered on current window)
		Fit all data into the visible window
		Goto the origin (time = 0)
		Lock the X-axis when zooming/scrolling/pan
		Lock the Y-axis when zooming/scrolling/pan
		Autoscroll the window to follow the video frames being decoded
		Move the window left/right/up/down
		Measure the values at the center of the '+'
		Zoom in/out - centered at the cursor

See *View buffer analysis..., page 7-151* for more detail.

Goto view, synchronize views

See *Synchronized views/navigating the views, page 7-6*.

Help Menu



This menu allows the user to select the following items:

Help topics

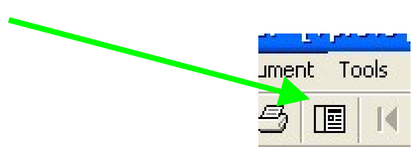
Help, organized into topics. This is the help that is displayed when the ‘F1’ key is pressed or  icon on the toolbar is clicked. This Help also includes the tutorials.

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 fully linked 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 works through the tutorials. The tutorials are also included in the standard Help (accessed via the ‘F1’ key), although the PDF version is easier to print out than the copy included in the standard help system.

License manager…

This displays the MTS4EA License manager, including:

- current license status information;
- other possible MTS4EA options that can be licensed;
- option to update the license key.

About …

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

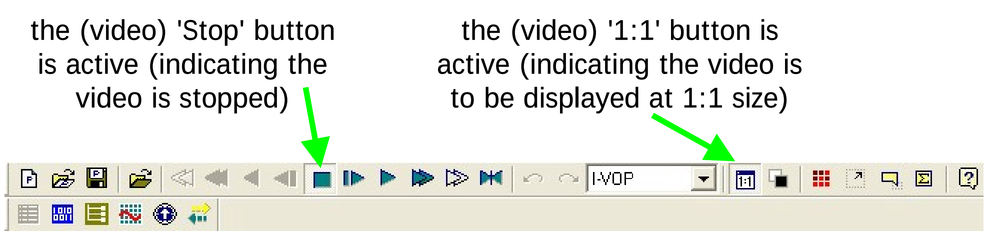
Disabled toolbar buttons

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, page 7-142*) the toolbar buttons concerned with the video files - such as ‘Play’, ‘Forward’, etc - will be unavailable as these are not applicable to the HexView.

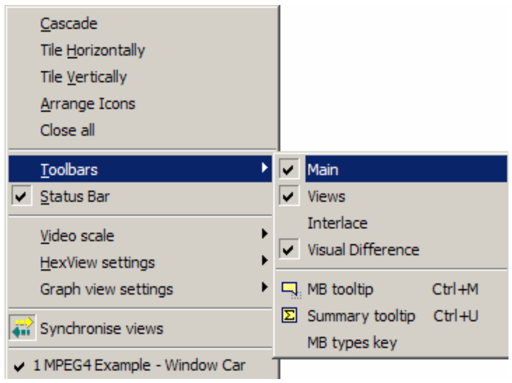
Active/inactive toolbar buttons



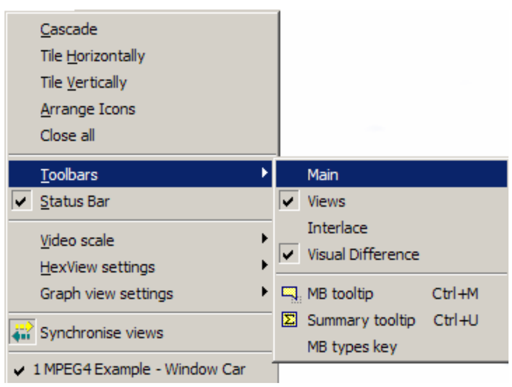
Displaying/hiding the Main toolbar

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 3 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	Manual section (see also…)
	O	Open video…	Open a video file	*Opening a video file, 7-5; Open video… Ctrl+O, 7-13;*
	P	Play forward	Play a video file (forwards)	*Play Ctrl+P, 7-38;*
	F	Fast forward	Fast forward a video file	*Fast forward Ctrl+F, 7-39;*
	B	Blind fast forward	Blind fast forward a video file (video blanked)	*Blind fast forward Ctrl+B, 7-39;*
	A	Pause/Step forward (Advance)	Pause a video file/advance by one frame	*Pause/Step forward Ctrl+A, 7-40;*
	S	Stop	Stop playing a video file	*Stop Ctrl+S, 7-40;*
		Pause on Frame	Pause on a specific frame number	*Pause On Frame , 7-43;*
	shift + P	Play backward	Play a video file backwards	*Reverse play Ctrl+Shift+P, 7-41;*
	shift + F	Fast backward	Fast backward a video file	*Fast backward Ctrl+Shift+F, 7-41;*
	shift + B	Blind fast backward	Blind fast backward a video file (video blanked)	*Blind fast backward Ctrl+Shift+B, 7-41;*
	shift + A	Pause/Step backward	Pause a video file/Step back by one frame	*Pause/Step backward Ctrl+Shift+A, 7-41;*
	K	Skip forward	Skip to the next I-frame/forward 'n' frames/forward 'n' seconds	*Skip forward Ctrl+K, 7-42;*
	shift + K	Skip backward	Skip back to the next I-frame/ back 'n' frames/back 'n' seconds	*Skip backward Ctrl+Shift+K, 7-42;*
	1	1:1	Set video scale: 'Best fit' or 1:1	*Video scale, 7-189;*
	W	Black/White	Set overlay digits to black/white	*Black/White , 7-98;*
	Y	MB Types	MacroBlock type overlay	*MB Types Ctrl+Y, 7-77;*
	E	Motion Vectors	Overlay the display of motion vectors	*Motion Vectors Ctrl+E, 7-80;*
	M	MacroBlock tooltip	Open/close the MacroBlock tooltip	*MB Tooltip Ctrl+M, 7-67;*

Toolbar icon	Ctrl +	Name	Description	Manual section (see also…)
	U	Summary tooltip	Open/close the Summary tooltip	*Summary Tooltip Ctrl+U, 7-57;*
	F1 **	Help	Go to Help topics	*Help Menu, 7-194*

** Note: 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.

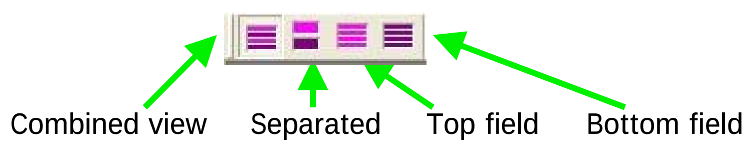
Toolbar icon	Ctrl +	Name	Description	Manual section (see also…)
	V	View current Trace	View the currently selected Trace file	*View trace… Ctrl+V, 7-136;*
	H	View stream hex	Open the current video stream in the HexView	*View stream hex… Ctrl+H, 7-142;*
	R	View file structure	Open the current video/container file and view the structure in navigable 'tree' form	*View file structure... Ctrl+R, 7-148;*
		View video buffer analysis	Graphs of VBV/VCV/VMV etc. (depends upon standard)	*View buffer analysis..., 7-151;*
		Video navigator	Show thumbnail viewer of each video frame and basic information	*Video navigator… , 7-49;*
		Synchronize views	Synchronize all open views	*Synchronized views/navigating the views, 7-6;*
		View fidelity analysis	Show fidelity analysis	*View fidelity analysis... , 7-163;*
		View alert log	Display log of alerts	*Alert log , 7-178;*

Context-sensitive Toolbars/Tooltips

These are toolbars and tooltips (or sometimes information windows) which 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 as described in *Summary and Macroblock Tooltips: moving, docking, scroll bars, page 7-53*.

Interlace toolbar



Only one of these 4 buttons is pushed in (active) at any time.

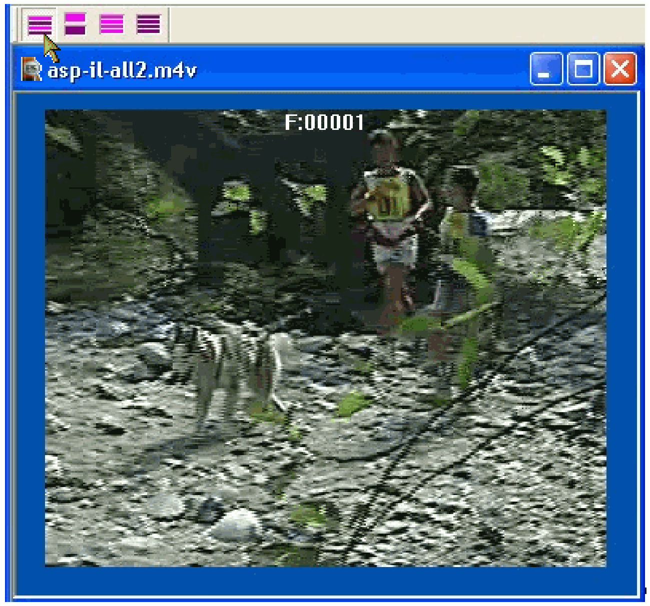
NOTE. *The Interlace toolbar only appears when a bitstream could have Interlace within it.*

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.

Examples of possible video streams where the Interlace toolbar could appear are (this is not a complete list):

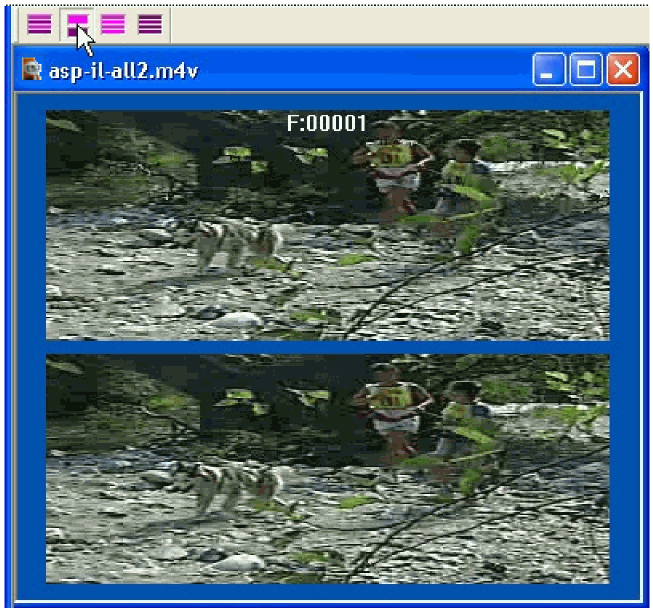
- H.264/AVC Extended Profile, Main Profile
- MPEG-4 Advanced Simple Profile, Levels 4 and 5
- MPEG-2 Main Profile, Main Level.

Combined 'frame' view. Both fields are shown together, as a single image, as below:



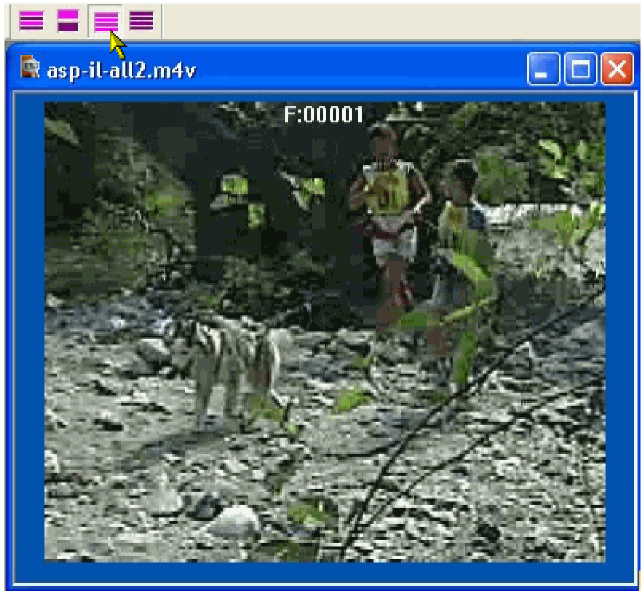
NOTE. *Some data and functions are not available when the video is displayed in this view: for example, the MacroBlock tooltip is empty and cannot be used (because the cursor would be spanning two different MacroBlocks: one in each field); motion vectors, MacroBlock Types and other overlaid data do not 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

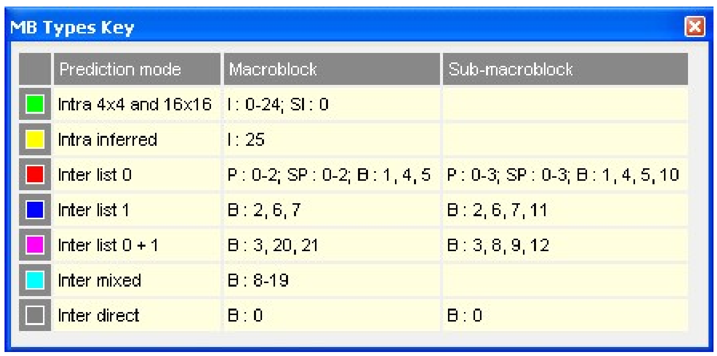
This is an information tooltip that appears when the MacroBlock Types overlay is switched on: its function is to explain the colors used.

This appears when the MB types overlay is switched on (the icon is active).

HINT. *To force undocking of the MB types color key tooltip, hold the <Ctrl> key while dragging with the mouse*

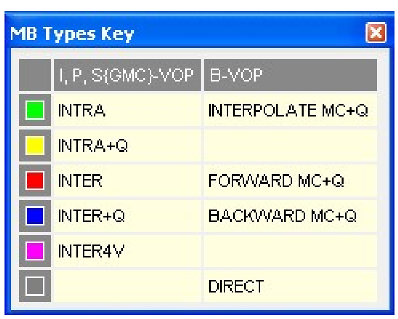
This tooltip can be closed by clicking on the 'X' at the top of the tooltip: to redisplay it click the MB types overlay icon off then on.

HINT. *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'.*



Prediction mode	Macroblock	Sub-macroblock
Intra 4x4 and 16x16	I : 0-24; SI : 0	
Intra inferred	I : 25	
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



I, P, S(GMC)-VOP	B-VOP
INTRA	INTERPOLATE MC+Q
INTRA+Q	
INTER	FORWARD MC+Q
INTER+Q	BACKWARD MC+Q
INTER4V	
	DIRECT

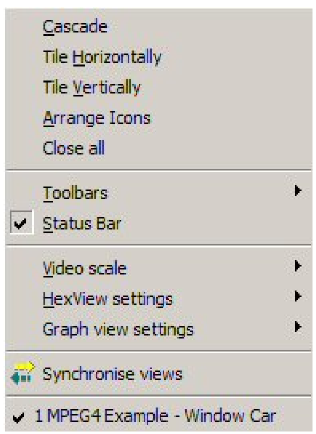
MPEG-4 example

Status Bar

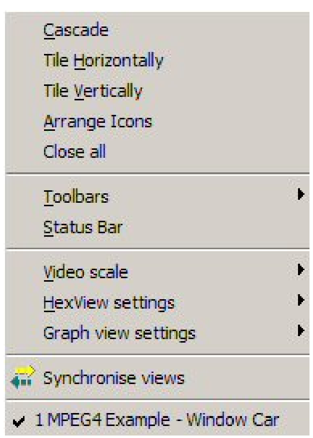
This provides:

- a one-sentence description of each menu item, as the mouse is moved over the menu item;
- the video 'play' mode (see *Video playing mode: restrictions, page 7-6*);
- context-sensitive information i.e. 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:



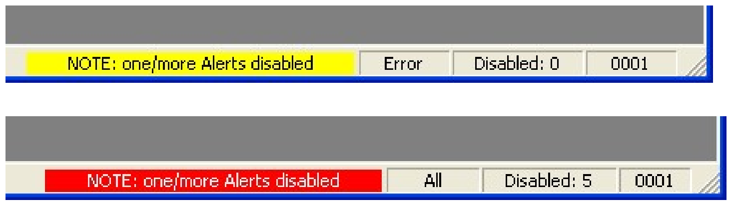
Status bar displayed



Status bar hidden

For more information on status bar indications of:

Alert levels, alert warnings, alerts disabled:

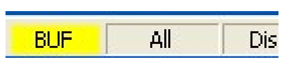


See *Status bar indication of Alert status, page 7-185*.

Frame range (in/out, etc.):

See *Frame range in/out indicator on status bar, page 7-91*.

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 *Buffer analysis controls: MPEG-4 and MPEG-2, page 7-158* and *Use custom buffer parameters: MPEG-4 and MPEG-2, page 7-159*.

'Ctrl' Shortcut Keys

All shortcut keys are accessed by holding the ‘Ctrl’ (Control) key and pressing the letter given below, EXCEPT the F1 key:

Ctrl +	Toolbar icon	Name	Description	Manual section (see also…)
mouse		Force un-dock	Force un-docking of a tooltip (e.g. Summary/MacroBlock)	-
1		1:1	Set video scale: 'Best fit' or 1:1	Video scale, 7-189;
A		Pause/Step forward (Advance)	Pause a video file/advance by one frame	Pause/Step forward Ctrl+A, 7-40;
shift+A		Pause/Step backward	Pause a video file/Step back by one frame	Pause/Step backward Ctrl+Shift+A, 7-41;
B		Blind fast forward	Blind fast forward a video file (video blanked)	Blind fast forward Ctrl+B, 7-39;
shift+B		Blind fast backward	Blind fast backward a video file (video blanked)	Blind fast backward Ctrl+Shift+B, 7-41;
E		Motion Vectors	Overlay the display of motion vectors	Motion Vectors Ctrl+E, 7-80;
F		Fast forward	Fast forward a video file	Fast forward Ctrl+F, 7-39;
shift+F		Fast backward	Fast backward a video file	Fast backward Ctrl+Shift+F, 7-41;
G		Graph enable	enable the Graph output	Graph enable… Ctrl+G, 7-122;
H		View hex	Open the current video stream in the HexView	View stream hex… Ctrl+H, 7-142;
K		Skip forwards	Skip forwards to the next frame type/no. of frames/no. of seconds	Skip forward Ctrl+K, 7-42;
shift+K		Skip backwards	Skip backwards to the next frame type/no. of frames/no. of seconds	Skip backward Ctrl+Shift+K, 7-42
M		MacroBlock tooltip	Open/close the MacroBlock tooltip	MB Tooltip Ctrl+M, 7-67;
O		Open video…	Open a video file	Open video… Ctrl+O, 7-13;
P		Play (forwards)	Play a video file (forwards)	Play Ctrl+P, 7-38;
shift+P		Play backwards	Play a video file (backwards)	Reverse play Ctrl+Shift+P, 7-41;
R		View file structure	View the structure of a video file	View file structure... Ctrl+R, 7-148;
S		Stop	Stop playing a video file	Stop Ctrl+S, 7-40;
T		Trace enable	enable the Trace output	Trace enable Ctrl+T, 7-102;

Ctrl +	Toolbar icon	Name	Description	Manual section (see also…)
U		Summary tooltip	Open/close the Summary tooltip	Summary Tooltip Ctrl+U, 7-57;
V		View current Trace	View the currently selected Trace file	View trace… Ctrl+V, 7-136;
W		Black/White	Set overlay digits to black/white	Black/White, 7-98;
Y		MB Types	MacroBlock type overlay	MB Types Ctrl+Y, 7-77;
F1 **		Help	Go to Help topics	Help Menu, 7-194;
F3 **		find next	in Trace and HexView only	View trace… Ctrl+V, 7-136 and View stream hex… Ctrl+H, 7-142
Shift+F3 **		find previous	in Trace and HexView only	View trace… Ctrl+V, 7-136 and View stream hex… Ctrl+H, 7-142

** Note: the 'F1' and '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 menu.

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/check mark 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.

HINT. *Don't forget 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, or
- 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 (i.e. including any file extension after the ".").

Using demonstration sequences in batch mode. To use batch mode with the demonstration tutorial video sequences provided on the ‘File’ menu, the filename for these sequences is specified below:

Demo sequence	Standard	Filename
H.264/AVC byte streams		
Neon Night	H.264/AVC Baseline Profile/Level 2	avc_1
Canary Wharf	H.264/AVC Extended Profile/Level 3	avc_2
Bus Junction	H.264/AVC Baseline Profile/Level 3	avc_3
Grenadier Guards	H.264/AVC Main Profile/Level 3	avc_4
MPEG-4 Elementary streams		
Woman Drinking	MPEG-4 Main Profile	mpg4_1
Train in Station	MPEG-4 Main Profile	mpg4_2
Space	MPEG-4 Simple Profile	mpg4_3
Man Walking	MPEG-4 Advanced Simple Profile	mp4asp_1
Synthetic	MPEG-4 Advanced Simple Profile	mp4asp_2
Window Car	MPEG-4 Advanced Simple Profile	mp4asp_3
H.263 streams		

Demo sequence	Standard	Filename
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
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

To use these demonstration sequences, the option '`-d`' is used - see *Command line/batch mode options, page 7-213*.

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	No. of frames
guards_yuv	H.264/AVC Grenadier Guards MPEG-2 Grenadier Guards	25	9
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

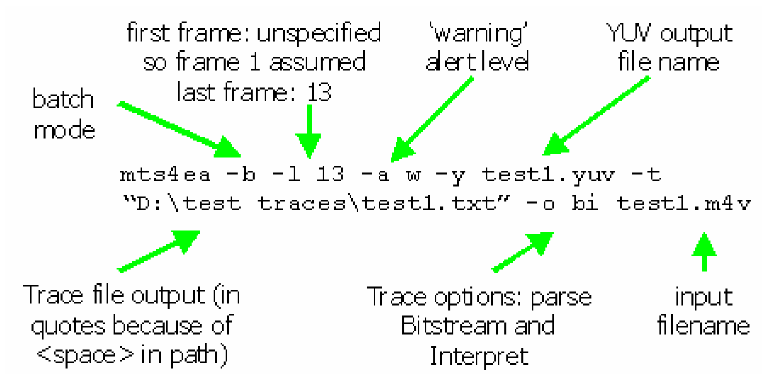
Option	Value (if any)	Description
-a	<alertlevel>	Alert level, which must be one of the following text strings: error warning fatal As with MTS4EA run in Windows mode, if the 'warning' alert level is on, trace outputs will also include 'error' and 'fatal' level alerts. 'fatal' level alerts are always 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 in *Specification of filename (input video file), page 7-211* (this filename is given at the end of the command line, like any other input filename).
-f	<firstframe>	integer value specifying the first frame to be used for the YUV and Trace outputs. If this option is not specified then frame number 1 (the first frame in the video sequence) is used.

Option	Value (if any)	Description
-h		displays help Window (MTS4EA opens after 'OK' is clicked).
-i	<track ID>	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 MBytes 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).
-o	<options>	the options are any combination of the following letters with no spaces between them: option trace output generated f Frame summary g GOB summary m MacroBlock summary b parse Bitstream d DCT level p Pixel level i Interpret
-r	<YUV file>	full name of the YUV reference file used for fidelity analysis. (see also the options -s, -u, -v).
-r	<size>	header skip of the YUV reference file used for fidelity analysis. (see also the options -r, -u, -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, -v).

Option	Value (if any)	Description
-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 Sum of Absolute Differences (see also the options -r, s, -u).
-w	<error_file>	output trace file with warnings/errors/fatal alerts only - no other trace information. <error_file> is the name of the file which holds the warnings/errors/fatal alerts. If there are no warnings/errors/fatal alerts then the <error_file> is zero length.
-y	<yuvfile>	full name of the YUV output file (format described in *Command line/batch mode options, page 7-213*).

Example command line.

An example of a valid command line is:



Format of YUV file output from batch mode.

NOTE: *the YUV output file in this section is the YUV output resulting from decoding the compressed video: this is a different file from the YUV reference file which is used when doing fidelity analysis.*

The YUV file output is ‘raw’ YUV with no headers of any kind: this is the same format as used by the Microsoft reference encoder reference [7] in *Standards References, page 5-14* and as used commonly by other programs:

- no headers of any kind (no file or frame headers)
- one byte per sample
- progressive scan (not interlaced)
- row raster order (top picture row first)
- planar YUV 4:2:0 sub-sampled i.e. 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.

.

Bitstream Syntax Debugging

Bitstream Syntax Debugging

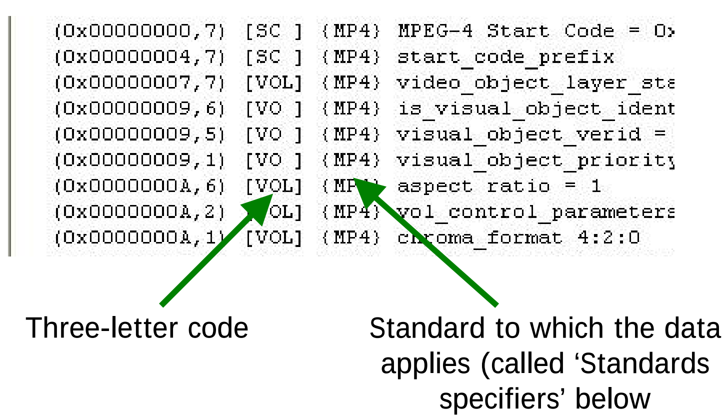
This section gives some tips and suggestions on debugging the syntax of video bitstreams syntax.

In order to understand these, you will need to have the relevant video standard(s) to hand.

NOTE. *This manual section applies to streams which 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:



Explanation of 3-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.

3-letter codes for H.264/AVC.

3-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
SEI	
BUP	buffering_period
DRR	dec_ref_pic_marking_repetition
FLP	filler_payload
FFF	full_frame_freeze
FFR	full_frame_freeze_release

3-letter Code	Syntactic Structure in H.264/AVC
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

Where appropriate these are paired together in the form:

[TTT:LLL]

where

- TTT indicates the top called syntactic structure and
- LLL indicates the lowest called syntactic structure

which are used to generate the data given in the Trace file.

3-letter codes for MPEG-4 and H.263.

3-letter Code	Name in H.263	Name in MPEG-4 Std. section
[SC]	Picture Start Code PSC	Start Code6.2, 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	macroblock6.2.6
[MOV]	MVD, MVD_{2-4}	motion_vector 6.2.6.2
[BLK]	Block layer	block 6.2.7
[inf]	Information about the internal state of the decoder or variables that may be described in the relevant standard	

3-letter codes for MPEG-2.

3-letter Code	Syntactic Structure in MPEG-2
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

3-letter Code	Syntactic Structure in MPEG-2
CTC	capture_timecode
EXD	extension_data
FCT	frame_or_field_capture_timestamp
MBK	macroblock
MBM	macroblock_modes
MVS	motion_vectors
NSC	next_start_code
PAD	padding_bytes
SLI	slice
USD	user_data
VEC	motion_vector
Extensions	
CRX	copyright_extension
CPX	camera_parameters_extension
ITU	itu_t_extension
PDX	picture_display_extension
PSS	picture_spatial_scalable_extension
PTS	picture_temporal_scalable_extension
QMX	quant_matrix_extension
SDX	sequence_display_extension
SSX	sequence_scalable_extension

Where appropriate these are grouped together in the form:

[TTT:MMM:LLL]

where

- TTT indicates the top called syntactic structure and
- MMM indicates the middle called syntactic structure(s) and
- LLL indicates the lowest called syntactic structure

which are used to generate the data given in the Trace file.

Explanation of 'Standards specifiers'

The following 'standards specifiers' are inserted into the Trace/Interpret, Alerts and (sometimes) Trace/Parse Bitstream outputs.

These are used to relevant where a particular standard applies uniquely to the data given.

NOTE. *The use of the Standards Specifiers is being removed from this and future versions of MTS4EA, because the standard concerned is stated at the start of the parse bitstream/interpret trace file and therefore it is unnecessary to have these indicators on every line. This applies to H.264/AVC and MPEG-2.*

Standard specifier	Standard	Explanation
{MP4}	MPEG-4	Applies ONLY to MPEG-4 If a Trace or Alert value has an {MP4} label and the video is not MPEG-4 then this indicates an error in the bitstream
{263}	H.263	Applies ONLY to H.263 baseline or H.263+ If a Trace or Alert value has an {263} label and the video is not H.263 then this indicates an error in the bitstream
{263+}	H.263+	Applies ONLY to H.263+. This also is used to indicate the use of more than one Annex in H.263+ If a Trace or Alert value has an {263+} label and the video is not H.263+ (for example H.263 baseline) then this indicates an error in the bitstream
{263 Annex L}	H.263+	Applies ONLY to the relevant Annex of H.263+ If this specifier is present in Trace or Alert outputs and the Annex concerned has not been used in the video then this indicates an error in the bitstream

Sometimes, the same bitstream field is used in H.263 and MPEG-4, but is given different names, and it is not necessarily possible for MTS4EA to produce different print-outs, for example with MPEG-4 Short Header, which is the same as H.263 baseline, where different mnemonics are used.

When this occurs, the two standards concerned will be listed in brackets, e.g. {MP4} or {263}.

Two examples of this are a section from a Trace/Interpret file (for MPEG-4 Short Header/H.263 baseline):

```
(0x00000002,1) [VPS] temporal_reference (MP4) / TREF (263) = 0; Cumul. skip = 255
(0x00000003,1) [VPS] 1-0-ssi-dci-fpfr-sf (MP4) / PTYPE (263) = 0x87
```

- In this situation if you are working with MPEG-4 the bitstream field at position 2, 1 is called 'temporal_reference' but 'TREF' (or 'TR') in H.263.
- Similarly, the next byte is called 'PTYPE' in H.263 (with the bit-fields listed as parts of PTYPE) but the bit fields are separately in MPEG-4.

Explanation of Bit/Bit Start

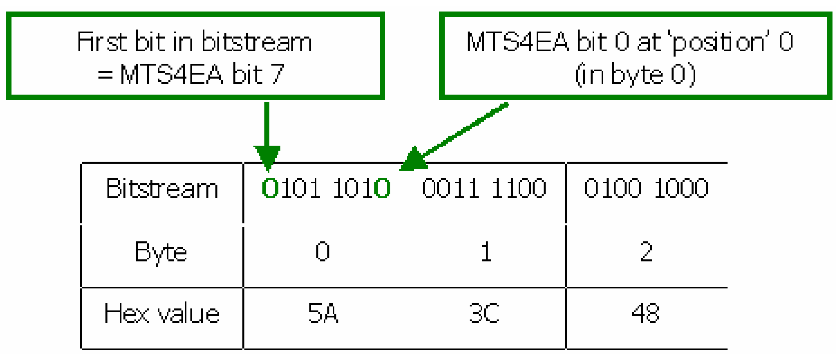
In each of the compression standards there are many bitstream fields which give values which are 'least significant bit first' ('lsbf') and many others which are 'most significant bit first' ('msbf'). There are also many others which are simply bit-patterns: they do not have a direct numeric value.

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:



Therefore a bitstream field where the:

- position is 1
- and the bit/bit start is 4
- and the length is 7 bits

covers the above bitstream:

Bitstream	0101 1010	0011 1100	0100 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 *Alerts Menu, page 7-166*); and
- the Trace/Parse Bitstream function (see *Trace enable Ctrl+T, page 7-102*); and
- the Trace/Interpret function (see *Trace enable Ctrl+T, page 7-102*).

Essentially, of the 3 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 *Bitstream Syntax Debugging using MTS4EA, page 8-9*);
- 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);
- You can 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)

HINT. *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 may 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, page 7-142*.

Overview

In general it is better to use the following order:

- Alerts; then
- Analysis/Trace/Parse Bitstream (with or without the HexView); then
- Analysis/Trace/Interpret.

However, for some debugging, it is useful to use Trace/Interpret first and then Parse Bitstream, 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 video...).

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 (i.e. byte address) and bit position given. (See *Explanation of Fatal/Error/Warning/Info display, page 7-167* 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, page 8-11*.

  d. If there are no Errors or Warnings on File/Open, then Play the video sequence and see if any Errors or Warnings occur.

  a. If no errors or warnings occur, then go to Trace/Parse Bitstream debugging, page 8-11.

5. If there are Errors and/or Warnings, then stop playing the video; switch **on** the Summary Tooltip (see *Summary Tooltip Ctrl+U, page 7-57*) and play the video again, until it stops at the first Error/Warning.

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, page 8-11)*.

Trace/Parse Bitstream debugging

Procedure.

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, page 7-102*).

2. Set the Trace filename and the maximum Trace File size (see *Trace enable Ctrl+T, page 7-102*).

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 then see a Trace out file similar to that given in *Trace enable Ctrl+T, page 7-102*.

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/Parse Bitstream debugging, *page 8*-11.

9. If there are Errors/Warnings 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 what is going on.

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.

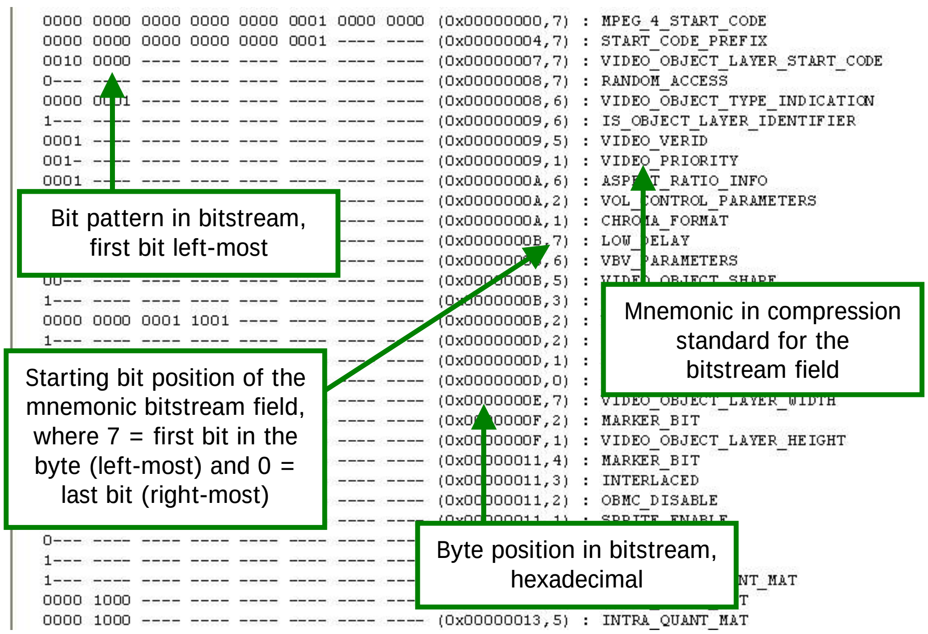
For example:

- 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 17^{th} bit after 'pei'=0 would not be set to 1).

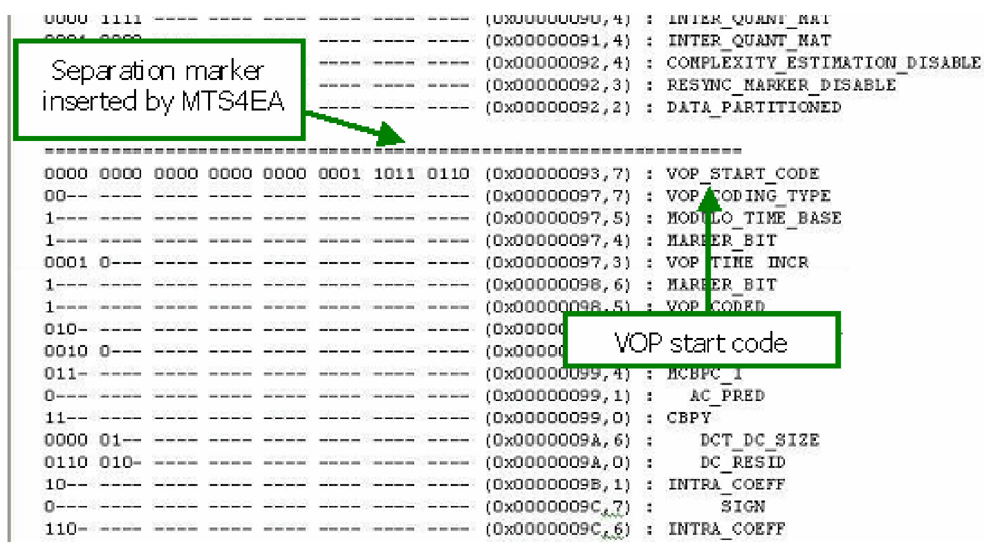
10. Check that there are no elements of a standard being used which should not be there: for example, if you are compressing data to H.263 baseline or MPEG-4 short_header, search for "{263+}" and "{263 Ann" [to find any H.263+ annexes used).
Search for the 'standards specifiers' used by MTS4EA, as given in section 0.

11. Check the 3-letter codes in the square brackets used by MTS4EA and reported in the trace output, such as "[SC]", as given in *Explanation of 3-letter codes, page 8-1*, and that the 3-letter code corresponds with the relevant standard section.

Parse Bitstream example outputs. See also section 0 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, where 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 best to use both Trace/Parse Bitstream and Trace/Interpret.

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, page 7-102*).
2. Set the Trace File name and the maximum Trace File size (see *Trace enable Ctrl+T, page 7-102*).
3. Set Interpret **on** and all other Trace functions **off**.
4. Put 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 *Trace enable Ctrl+T, page 7-102*.
7. Search for 'Errors' and 'Warnings' (using the 'Find next' button in the 'View trace...' window).
8. If there are Errors/Warnings/Out of Sync 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 what is going on.

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;

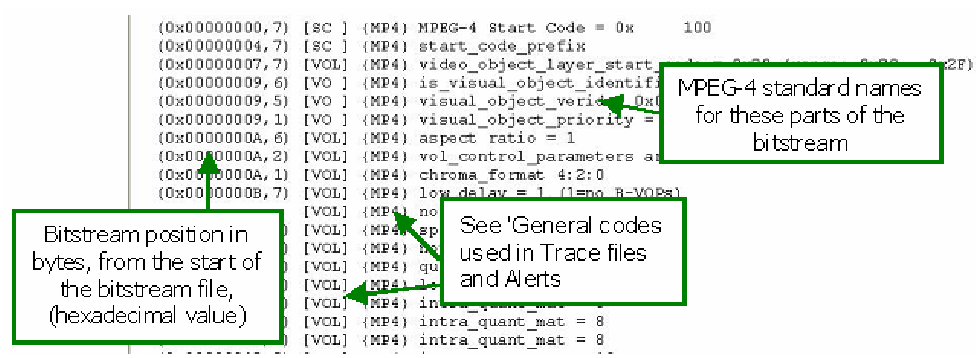
- therefore if ‘pei’ was set to 0 by accident, but there was a ‘psupp’ field inserted set to all zeroes, the bit pattern following ‘pei’ would be:
  0000 0000 0000 0000 0000 0000 1
- this would appear to be an error in the gob_resync_marker (that is because the 17th bit after ‘pei’=0 would not be set 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].

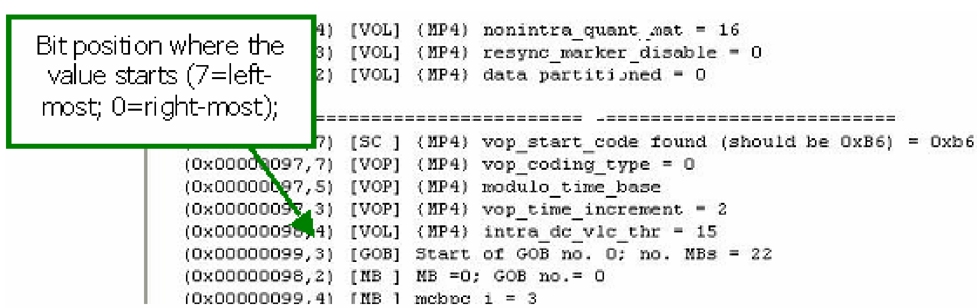
Search for the ‘standards specifiers’ used by MTS4EA, as given in *Explanation of ‘Standards specifiers’, page 8-6*.

10. Check the 3-letter codes in the square brackets used by MTS4EA and reported in the trace output, such as “[SC]”, as given in *Explanation of 3-letter codes, page 8-1*, and that the 3-letter code corresponds with the relevant standard section.

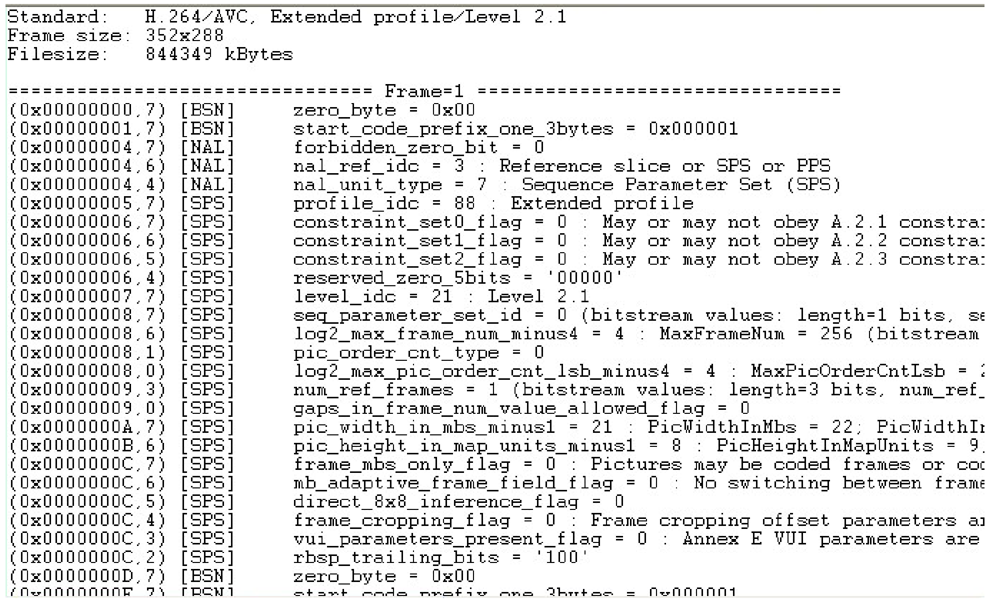
Interpret example outputs. See also *Trace enable Ctrl+T, page 7-102* for more information.



MPEG-4 Example, at start of bitstream

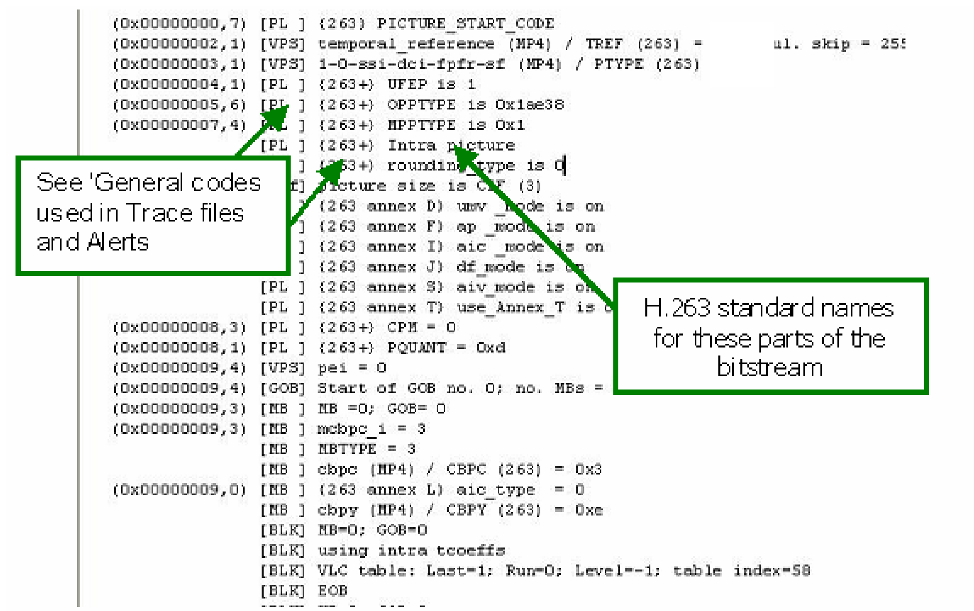


MPEG-4 Example, at start of VOP



```
Standard:   H.264/AVC, Extended profile/Level 2.1
Frame size: 352x288
Filesize:   844349 kBytes

================================ Frame=1 ================================
(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 constra:
(0x00000006,6) [SPS]     constraint_set1_flag = 0 : May or may not obey A.2.2 constra:
(0x00000006,5) [SPS]     constraint_set2_flag = 0 : May or may not obey A.2.3 constra:
(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 = :
(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; PicWidthI
(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 a
(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

Purpose of MTS4EA Decoder Plugins

The MTS4EA Decoder Plugins allow the advanced user to substitute various elements of the 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, page 7-210*) - 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 *Decoder Plugins Provided, page A-4* 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 *Decoder Plugins Provided, page A-4* for an example.
- sometimes it is helpful to be able to use the MTS4EA function within a user's own decoder, to guarantee that numerically-identical results are output by the 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 MTS4EA 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 MTS4EA 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 MTS4EA 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 which 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`'.

Development Tools, 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 Prerequisites, section 3.

NOTE. *No other development tools or variants to these is 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 the 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 which are in the MTS4EA executable folder are the 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*, section 5) 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*, section 5), although it is (theoretically) ‘normative’ is one such implementation which does not actually implement the quarter sample calculations in accordance with the standard.

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*, section 5) differs from the MPEG-4 standard (reference [1] in *Standards References*, section 5) 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 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, as per Reference [7] in *Standards References*, section 5
- MPEG-4 Normative ISO bitstreams: dated 05/11/2001, as per Reference [9] in *Standards References*, section 5
- 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*, section 5

"-n/a-" = "not applicable"

Bitstreams: Normative ISO

Stream name	Decodes with MTS4EA?	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 MTS4EA?	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 MTS4EA?	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 MTS4EA?	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 MTS4EA?	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 MTS4EA?	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). MTS4EA 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 MTS4EA?	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 MTS4EA?	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 more easy 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

B

C

D

O

P

T

V