

Matrox DSX

Release Notes (Windows)
Version 9.4

April 2, 2014



Y11301-401-0940

Trademarks

Matrox Electronic Systems Ltd.	Matrox [®] , DSX [™] , DSX.sdk [™] , DSX.utils [™] , FlexEffects.sdk [™] , X.264 [™] , X.264io [™] , X.AVCio [™] , X.mio [™]
Apple Inc.	Apple [®] , QuickTime [®]
Avid Technology, Inc.	Avid [®] , DNxHD [®] , Media Composer [®]
Dolby Laboratories Inc.	Dolby [®]
Harris Broadcast	Harris [®] , NEXIO [®]
Intel Corporation	Intel [®]
Microsoft Corporation.....	Microsoft [®] , DirectX [®] , Visual Studio [®] , Windows [®] , Windows Media [®]
Minnetonka Audio Software, Inc.	SurCode [™]
Panasonic (Matsushita Electric Industrial Co.).....	Panasonic [®] , AVC-Intra [™]
Sony Corporation	Sony [®] , XDCAM [®]

All other nationally and internationally recognized trademarks and tradenames are hereby acknowledged.

Copyright © 2014 Matrox Electronic Systems Ltd. All rights reserved.

Disclaimer Matrox Electronic Systems Ltd. reserves the right to make changes in specifications at any time and without notice. The information provided by this document is believed to be accurate and reliable. However, no responsibility is assumed by Matrox Electronic Systems Ltd. for its use; nor for any infringements of patents or other rights of third parties resulting from its use. No license is granted under any patents or patent rights of Matrox Electronic Systems Ltd.

Unauthorized recording or use of broadcast television programming, video tape, or other copyrighted material may violate copyright laws. Matrox Electronic Systems Ltd. assumes no responsibility for the illegal duplication, use, or other acts that infringe on the rights of copyright owners.

Matrox Electronic Systems Ltd.
1055 St. Regis Blvd., Dorval, Quebec, Canada H9P 2T4
Tel: (514) 685-2630 Fax: (514) 685-2853 World Wide Web: www.matrox.com

Contents

About these release notes	1
New features	1
New hardware features.....	1
New software features	1
Fixes	2
Fix for P2 MXF capture	2
FPGA fixes for X.mio2, X.mio2 Plus, DSX LE3, and X.RIO module	3
Important information about removal of static libraries	3
Required GUIDs for using the Flex File Writer	4
Enabling H.264/AVC encoding using Intel QSV	5
Selecting audio streams for playing back .mov and .mxf files	5
Supported operating systems	6
Development platform	6
Supported versions of Apple QuickTime and Avid DNxHD	6
Known issues	7
Software issues	7
Installation issues	8
Downloading this release	9

Your notes

About these release notes

This document lists new features, fixes, known issues, and other important information applicable to the Matrox DSX version 9.4 release.

New features

This release provides the following new features:

New hardware features

- Support for the new DSX LE3 4K card (DSX LE3/4KO). This model of the DSX LE3 card supports SDI video output ranging from SD up to 4K at 60 fps with 3G Level A and Level B.

New software features

- Support for 8-bit 4:2:0 H.264/AVC multiple-stream encoding in 32- and 64-bit applications, which can be accelerated when run on a system with Intel Quick Sync Video (QSV). This support includes capture to resolutions ranging from proxy up to HD, ancillary data capture, and capture to *.264*, *.mov*, and *.mp4* files with AAC audio. For details on how to enable hardware-accelerated encoding, see “[Enabling H.264/AVC encoding using Intel QSV](#)” on page 5.



Note This feature is not supported on Windows XP.

- Support for playing back Harris NEXIO LXF files.
- Support for capturing to Avid DNxHD OP1a *.mxf* files.¹
- Support for 1080p @ 50 and 59.94 fps capture and playback with Avid DNxHD.¹
- Support for MPEG-2 transport streams with H.264/AVC video and AAC audio containing SCTE 35 signaling protocols, closed caption, AFD, bar data, and time code information. This feature requires the Matrox H.264/AVC hardware codec.
- Support for 2K, QFHD, and 4K resolutions with Apple ProRes.²
- Integration of the Minnetonka SurCode for Dolby E codec with Matrox DSX.sdk.³
- Support for 16- and 24-bit audio with most MXF file formats.
- Improved support for 3G Level B SDI input/output on X.mio2.

¹ A license agreement with Avid is required in order to support DNxHD clips in your application. For more information about using DNxHD, please contact your Matrox sales representative. To acquire the DNxHD license agreement, please contact Avid Technology at avidDNxHD@avid.com.

² A license agreement with Apple is required in order to support ProRes clips in your application. For more information about using Apple ProRes, please contact your Matrox sales representative.

³ A license agreement with Minnetonka Audio Software is required in order to use the SurCode for Dolby E codec in your application. For details on how to obtain the required license agreement, please contact Minnetonka Audio Software at contact@minnetonkaaudio.com.

- Support for playing back *.mov* and *.mxf* files with more than 16 audio tracks (see “[Selecting audio streams for playing back .mov and .mxf files](#)” on page 5).
- Support for selecting the time code location for playing back *.mxf* files.
- Support for changing the initial time code for capturing *.mxf*, *.gxf*, and *.mov* files.
- New *bEnabledHighQualityXDCAM* encoding parameter in the **SMvMpegStreamEncoderSettings** structure that can be used to improve the overall compression quality of XDCAM HD/HD422 encoding. When enabled, the compression quality is improved by using more CPU processing, and applying internal hardcoded settings. For details, see the **SMvMpegStreamEncoderSettings** structure in the *Matrox DSX.sdk Programming Reference Library* online help.
- Support for Windows 8.1 64-bit and Windows Server 2012 R2.
- New sample applications:
 - *mvsTransportStreamWriter* Demonstrates how to write MPEG-2 transport stream packets to disk as a *.ts* file, either by transcoding or live capture.
 - *mvFlexDolbyETesterApp* Demonstrates how to capture and play back Dolby E audio streams.
 - *mvs2K4KCapture* Demonstrates how to capture to 2K, QFHD, and 4K files using X.mio2 and DSX LE3 cards that have four SDI inputs.
- General performance improvements.

Fixes

This release includes the following fixes:

Fix for P2 MXF capture

- In order to properly capture P2 MXF files that are compatible with Avid Media Composer, you must use the updated **SMvFlexWriterMxfP2CaptureExtraInfo** structure.

```
struct SMvFlexWriterMxfP2CaptureExtraInfo
{
    // Structure size in bytes.
    uint32_t size;
    // The total number of mono audio tracks to be captured.
    unsigned long ulTotalAudioTrackNums;
    // The sampling rate (samples per second) for the capture.
    unsigned long ulSamplesPerSec;
    // The ID of the audio track to be captured.
    unsigned long ulAudioTrackId;
    // The 16-byte UID of the Material Package.
    GUID guidMaterialPackageUID;
    // The 16-byte UID for the video and audio of the Source Package
    // in an array containing elements that are defined as follows:
    //     [0] for video.
```

```

//      [1] and [2] for two audio tracks.
//      [1], [2], [3], and [4] for four audio tracks.
//      [1], [2], ... [8] for eight audio tracks.
GUID aguidSourcePackageUID[33];
};

```

To create the 16-byte GUIDs, use the Windows **CoCreateGuid()** method.

FPGA fixes for X.mio2, X.mio2 Plus, DSX LE3, and X.RIO module

- Fixed noisy audio input issue that seldom occurred during a long capture session.
- Fixed issue in passthrough session that would momentarily corrupt the output audio after switching the input format many times.
- Fixed issue that caused the SDI output signal to occasionally be lost after powering up the card.
- Fixed blending issue that sometimes caused the luma to exceed the maximum allowable value.
- Fixed possible VANC packets masking issue for 0×04 values.
- Fixed some effect control issues on luminance key (input and output).
- Fixed passthrough issue that reverses the output colors after genlock source is interrupted (X.RIO module only).

Important information about removal of static libraries

The sample applications distributed with Matrox DSX.sdk link to a certain number of static libraries that are provided by Matrox for various helper mechanisms, and in some cases for basic functionality of the SDK (such as for GUID definitions). In an effort to reduce dependencies between the compiler and libraries that Matrox uses to build the SDK and the compiler that clients use to build their own applications, Matrox has started moving some class and function definitions that used to be distributed as static libraries to header files.

While care was taken to keep modifications that affect clients to a minimum, there are a few things some clients will need to do in order to build their applications with this release of the SDK.

- The following static libraries are no longer distributed with the SDK:
 - *mvCentralManager.lib*
 - *mvCommonMechanisms.lib*
 - *mvManagerTools.lib*
 - *mvdAPI.lib*

- *myfFlex3DDeviceAPI.lib*
- *myfUtility.lib*
- *mvSettingsManager.lib*

All functionality previously distributed in these libraries has been moved to their respective header files. If you are linking to these libraries, you will have to stop doing so by removing them from the Visual Studio project property page, under “Configuration Properties/Linker/Input/Additional Dependencies.” If you don’t do this, the linking stage will fail with a LNK1104 linker error.

- The **CMvPunctualEventObserver** class has been replaced by the **CMvPunctualEventWatcher** class that is now distributed with the SDK in the *mvPunctualEventWatcher.h* file (from the **mvsCommon** library). Clients that were previously using it must now include the new header file and rename occurrences of the class to the new one. None of the class methods have been changed.
- The **NMvfFlexEngineHelper** namespace has been replaced by the **NMvfFlexEngineCopyHelper** namespace that is now distributed with the SDK in the *myfFlexEngineCopyHelper.h* file (from the **mvsCommon** library). Clients that were previously using it must now include the new header file and rename occurrences of the namespace to the new one. None of the namespace functions have been changed.

Required GUIDs for using the Flex File Writer

With this release of Matrox DSX.sdk, when you specify the surface format for writing to a file using the Flex File Writer, you must also specify the GUID for the codec to be used to create the file. You do this by setting *guidCodec* in the **SMvFlexWriterVideoParameters** structure to the GUID for the codec that you want to use. For all formats except H.264/AVC, you can set *guidCodec* to *GUID_NULL*. If you want to encode to an H.264/AVC format, you must specify the appropriate GUID as follows:

Codec/encoder name	GUID	Supported surface format
Matrox H.264/AVC hardware codec (on X.AVCio, X.264, and X.264io only)	<i>CLSID_MvfAVCH264HardwareCodec</i>	<i>keMvSurfaceFormatAVC</i>
Matrox H.264/AVC software encoder	<i>CLSID_Mvch264SWEncoder</i>	<i>keMvSurfaceFormatAVC</i>
Matrox H.264/AVC QSV encoder (requires Intel Quick Sync Video)	<i>CLSID_Mvch264QSVEncoder</i>	<i>keMvSurfaceFormatAVC</i>

Enabling H.264/AVC encoding using Intel QSV

Matrox DSX.sdk supports H.264/AVC encoding using Intel Quick Sync Video (QSV). To use this feature, your computer must have an Intel processor with integrated QSV. To enable H.264/AVC encoding using QSV, do the following:

- 1 Update your computer with the latest Intel graphics driver at <http://downloadcenter.intel.com/>
- 2 On a Windows 7 system, if you want to work with two graphics cards (the onboard graphics and another card), do the following:
 - a Connect a monitor to each card and set the onboard graphics as the main display for your computer.
 - b Modify your computer's BIOS setting to activate multiple graphics cards.

Selecting audio streams for playing back .mov and .mxf files

To select the audio streams that you want to play back for .mov and .mxf files, call **IMvAVInterleavedPlayListElement** or **IMvAudioPlayListElement** using **SetAudioStreamIdentifier**(*in_ui32Index*, *in_ui64AudioStreamIdentifier*), where *in_ui32Index* is the index value for the group of audio streams, and *in_ui64AudioStreamIdentifier* specifies the particular streams to be played back.

Each stream represents one audio track, and can contain 1, 2, 4, 8, or 16 channels. Streams are indexed in groups of 64 (*in_ui32Index* is 0 for streams 0 to 63, 1 for streams 64 to 127, and so on). For example, to play back streams 0, 1, 2, and 4, call **SetAudioStreamIdentifier**(0, 0x00000017), and to play back streams 64 and 68, call **SetAudioStreamIdentifier**(1, 0x00000011).

Remarks

- At present, the number of audio streams must be less than 64. Therefore, *in_ui32Index* must be set to 0.
- The **GetNumberOfAudioStreamIdentifier()** and **GetAudioStreamIdentifier()** methods in **IMvAVInterleavedPlayListElement** and **IMvAudioPlayListElement** are not fully implemented yet.
- In the **SMvAudioVideoFileInfo** structure, *ulAudioStreams* indicates how many audio streams are in the file, and *sWaveInfo::eChannelType* indicates how many audio channels are in a stream.
- The number of channels per stream must be the same for all audio streams in the file.
- The minimum selection is one stream. You cannot play back specific channels from a stream.

- The total number of channels that can be played back is 2, 4, 8, or 16.
- The *MvfFlexChannelTester* sample application demonstrates how to use this feature.

Supported operating systems

It's highly recommended that you use one of the following Microsoft operating systems with this release:

- Windows 8.1 64-bit
- Windows Server 2012 R2
- Windows 7 64-bit with Service Pack 1
- Windows Server 2008 R2 64-bit with Service Pack 1

You should also develop 64-bit applications to take the best advantage of these 64-bit operating systems.

The following Microsoft 32-bit operating systems are also supported:

- Windows 7 32-bit with Service Pack 1
- Windows XP 32-bit with Service Pack 3

Development platform

This release supports the following development platform:

- Microsoft Visual Studio 2010 with Service Pack 1

Supported versions of Apple QuickTime and Avid DNxHD

This release supports the following versions of Apple QuickTime and Avid DNxHD:

- Apple QuickTime version 7.6. Be aware that QuickTime version 7.6.2 or later can cause capture and playback issues with the Matrox DSX.sdk proxy support.
- Avid DNxHD version 1.4.2. This version of the DNxHD codecs provides the best results for multi-threaded processing on systems that have more than 16 cores.

Known issues

Software issues

- Files captured using the Matrox H.264/AVC encoder with Intel QSV don't play back correctly using Apple QuickTime or a proxy session in the *MvfFlexChannelTester* sample application. (Ref# VPG-7441)
- When using the Minnetonka SurCode for Dolby E codec with Matrox DSX.sdk, selecting/changing the setting to configure the .wav file to anything other than 24-bit is not supported. (Ref# VPG-7172)
- On X.mio2 Plus, the Matrox Ancillary Data Format (specified with the surface format *keMvSurfaceFormatMatroxAncillaryData*) is not supported when an input or output stream in interlaced or PsF mode is connected to a node in progressive scan mode. In this case, only the legacy VANC (or VBI) surfaces stored as YUYV 4:2:2 10-bit data are supported. (Ref# VPG-4079)
- When you click on a link in the DSX.sdk online help, such as a link to a method in a table under **Public Methods**, you may get a message that the requested content can't be found on your computer. If this happens, click the link to the topic that you want in the navigation pane (table of contents) on the left. (Ref# VPG-3644)
- When performing a PiP session in *MvfFlexChannelTester* using 1080PsF video, you must set the Streaming Size to 6 and the Drive access to 12. (Ref# 35683)
- The *MvfFlexChannelTester* Playback session does not support mono .wma and .mp3 files. (Ref# 42096)
- Passing a name that's longer than 16 characters in the **CMvMutex** constructor will cause an unhandled exception error at execution time if the **CMvMutex** caller has been compiled in debug. (Ref# 42816)
- When performing a proxy capture using a QuickTime or Microsoft codec, the codec may accumulate data because it is not able to process all the frames in real time. Therefore, when you stop a proxy capture it may take longer than usual for the codec to close because it must process all accumulated frames before exiting. (Ref# 42935)
- To compile the *MvfFlexChannelTester* sample application, you must have the June 2010 DirectX SDK, available from the Microsoft website at <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=6812> (Ref# 44584)
- When using *MvfFlexChannelTester* to play back AVC-Intra or H.264/AVC files, you must set the Streaming Size to 16. (Ref# 47978)
- If you want to mix 8-bit and 10-bit files in the same playlist, such as 8-bit H.264/AVC files and 10-bit AVC-Intra files, you must add an 8-bit file first to set the playlist to 8-bit. Subsequent 10-bit files added to the playlist will then be converted to 8-bit. Adding a 10-bit file to the playlist first will set the playlist to

10-bit, and therefore prevent you from adding subsequent 8-bit files because the Flex File Reader does not support playing back 8-bit files in a 10-bit playlist. (Ref# 48690)

- Fast-motion playback of H.264/AVC *.mov* files is a very CPU-intensive operation. You may therefore experience dropped frames when performing fast-motion playback of these files on certain systems. (Ref# 49756)
- On a Windows Server system, Windows Media components are not installed by default. Therefore, you may experience problems trying to perform proxy captures to Windows Media (*.wmv* and *.asf*) files. If this happens, add “Desktop experience” to your Windows features as explained in your Windows documentation. (Ref# 53767)
- On a Windows 32-bit system, when using one instance of *MvfFlexChannelTester* to capture or play back multiple HD streams, the memory usage can become very high and cause the application to become unstable (the process has reached its maximum memory usage). It’s recommended that you use several instances of *MvfFlexChannelTester* if you want to work with several HD streams. As well, decreasing the Drive access size from 30 to 15 can help in the various sessions.
- When using the *MvfFlexChannelTester* New Layer session on X.mio2, sending many layers to the onboard scaler can cause dropped frames if the scaling engine’s performance is at its limit. For each layer, you can select an option to skip the scaling engine.
- Using the QuickTime Runtime Component to read/write *.mov* files with Matrox DSX.sdk is not supported on a Windows Server system. For details on QuickTime system requirements, see the QuickTime Pro technical specifications on the Apple website at <http://support.apple.com/kb/sp521>.

Installation issues

- On a Windows 8.1 or Windows Server 2012 system, the tiles for uninstalling Matrox DSX.sdk and Matrox DSX.utils may not be added to the Apps view. If this occurs, you can uninstall Matrox DSX.sdk or Matrox DSX.utils as follows:
 - In Control Panel**, switch to **Small icons view**, then choose **Program and Features**.
 - Choose **Matrox DSX.sdk** or **Matrox DSX.utils**.
 - Click **Uninstall**, then click **Next**.
 After Matrox DSX.sdk or Matrox DSX.utils has been successfully removed from your computer, click **OK**. (Ref# VPG-7045 and VPG-6091)
- You may experience issues installing DSX.utils over a network connection on systems that have certain third-party storage adapters. To avoid issues, install DSX.utils from the installation DVD, or copy the DSX.utils files to the system’s hard drive to perform the installation. (Ref# 49135)

- When you uninstall the Matrox DSX.sdk and choose to delete folders that contain additional content (such as folders created when you compile a sample application), the files in the folders will be deleted, but not the folders. You'll therefore have to delete these folders manually. (Ref# 51258)
- When you install the Matrox DSX.sdk on a Windows 64-bit system, the files are installed to the *\Program Files (x86)\Matrox DSX.sdk* folder by default. This is the default folder where Windows installs 32-bit applications on a 64-bit system. If you have problems compiling your applications because of the "(x86)" in the folder name, uninstall the Matrox DSX.sdk and re-install it to the *\Program Files\Matrox DSX.sdk* folder by changing the default folder name during the installation. (Ref# 51902)
- When installing Matrox DSX.utils, you may receive messages that some device drivers have not been installed successfully. You can ignore these messages, as the device drivers will be successfully installed once the DSX.utils installation is complete. (Ref# 54034)

Downloading this release

All details on recommended motherboards and systems can be found on the Matrox Private Developer Support website at: www.matrox.com/video/support/developer/private_support/login.cfm. It is highly recommended that you validate that a system meets your specific OEM requirements before selecting a system for use with your solution.

You can download the 9.4 release of DSX.utils, DSX.sdk, and FlexEffects.sdk at:

ftp://vid001@video.matrox.com/DSX/9409028_DSX.sdk_9.4_Release

Your notes



www.matrox.com/video