

Media Reactor SDK

Version 1.1



**DRASTIC
TECHNOLOGIES**

12 Drummond Street, suite 2
Eau Claire, ON, M9V 1Y8, Canada
Phone: (416) 255-5618
Fax: (416) 255-8788
Website: www.drastictech.com

Copyright (C) 2001 Drastic Technologies Ltd.
All Rights Reserved.

Information in this document is subject to change without notice and does not represent a commitment on the part of Drastic Technologies Ltd. The software described in this document is furnished under a license agreement or non-disclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or non-disclosure agreement. The licensee may make one copy of the software for backup purposes. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the licensee's personal use, without the express written permission of Drastic Technologies Ltd.

2001 Drastic Technologies Ltd. All rights reserved.
Printed in Canada

Any particular application may or may not take advantage of the MediaReactor features described within this document. Drastic Technologies Ltd. reserves the right to make changes in the specifications at any time and without notice. The information provided herein is believed to be accurate and reliable. However, no responsibility is assumed by Drastic Technologies Ltd. for its use; nor for any infringements of patents or other rights from its use. No license is granted under any patents or patent rights of Drastic Technologies Ltd. Drastic Technologies Ltd. makes no warranties, express or implied, with respect to the performance of third party products described herein.

Drastic Technologies Ltd. and MediaReactor are trademarks of Drastic Technologies Ltd. Other product names mentioned in this document may be registered trademarks or trademarks of other companies.

Contents

1	INTRODUCTION.....	1
2	INSTALLATION GUIDE.....	2
2.1	REQUIREMENTS.....	2
2.2	INSTALLING THE SOFTWARE.....	3
2.2.1	<i>MediaReactor Installation</i>	3
2.2.2	<i>License Authorization</i>	3
2.2.3	<i>Software License Issues</i>	4
2.2.4	<i>Sample Projects</i>	5
2.2.5	<i>Third Party Software Installation</i>	5
3	QUICK START.....	6
3.1	EXAMPLES.....	6
3.1.1	<i>Visual Basic – DTMRX_StepOne</i>	6
3.1.2	<i>Visual Basic – DTMRX_StepTwo</i>	6
3.1.3	<i>Visual Basic – DTMRXTest</i>	6
3.1.4	<i>Visual C - DTMXMFC</i>	6
4	REFERENCE GUIDE.....	7
4.1	FUNCTIONS.....	7
4.1.1	<i>Function Group: Utility</i>	7
4.1.2	<i>Function Group: File Info/Preview</i>	7
4.1.3	<i>Function Group: Get Available Functions</i>	9
4.1.4	<i>Function Group: Set File Output</i>	9
4.1.5	<i>Function Group: Set Video Output</i>	9
4.1.6	<i>Function Group: Set Audio Output</i>	10
4.1.7	<i>Function Group: Set Input And Output Files</i>	11
4.1.8	<i>Function Group: Process</i>	11
4.1.9	<i>Function Group: Translation</i>	12
4.1.10	<i>Function Group: Advanced</i>	13
4.2	MEDIA FORMATS.....	13
4.2.1	<i>Video for Windows (AVI)</i>	13
4.2.2	<i>Open Media Framework Interchange (OMF)</i>	14
4.2.3	<i>QuickTime (MOV)</i>	16
4.2.4	<i>Audio Files</i>	18
4.2.5	<i>Still Image Files</i>	18

1 Introduction

This is a manual for MediaReactor SDK Version 1.1 of file conversion software from Drastic Technologies Ltd.

The MediaReactor SDK provides an ActiveX interface to the MediaReactor file handling functionality. This ActiveX provides file information on source media files, RGBA and PCM audio reading for thumbnail and preview generation, as well as translation from supported media files and plug-ins to supported output types.

This manual is organized into the following sections:

- ◆ The Installation Guide provides instructions on how to install the software, as well as information on specific configuration issues.
- ◆ The Quick Start section of this manual is provided to give an overview as well as a tutorial on using the software so you may get started right away.
- ◆ The Reference section provides a list of the SDK functions and digital media files that the MediaReactor can process.

Document Conventions

The following conventions are used throughout this manual:

TYPE STYLE	USED FOR
Bold	Bold type is used to identify a label on a control in the interface.
<i>Italic</i>	<i>Italic</i> is used to identify words which represent concepts as used within MediaReactor

2 Installation Guide

The Installation Guide is divided into the following parts:

The Requirements section details the minimum hardware configuration you must have to install MediaReactor. Confirm that the system you wish to install the MediaReactor software on meets these requirements.

The Installing the Software section offers step by step instructions on installing the MediaReactor software.

2.1 Requirements

MediaReactor runs on the Microsoft Windows NT Version 4 or Windows 2000 operating systems. When using the NT Operating System, make sure that Service Pack 5 or greater has been installed. Apple's QuickTime Version 4 or greater must be installed on your system.

The SDK has been tested with Visual Studio 6 with Service Pack 3 installed. Any other development environment that can use ActiveX components should work with the SDK.

The minimum hardware configuration to run this software is:

- Pentium III processor, 450MHz
- 128 MB of memory (RAM)
- 20MB hard drive space required for installation
- CD-ROM drive
- SVGA or higher-resolution video adapter (minimum display resolution of 1024 x 768 full color) and a compatible monitor
- A Windows-compatible sound card (if it is necessary to preview sound files)
- Mouse and keyboard

2.2 Installing the Software

The MediaReactor SDK will install the ActiveX components, their supporting files and zipped versions of the sample projects.

2.2.1 MediaReactor Installation

Confirm that the computer is configured according to the minimum system requirements. Turn the computer on. Make sure that you are logged in as a User with Administrative privileges.

1. Make sure your system clock is properly set. This may be accomplished through the Control Panel, in Date/Time.
2. Place the CD in the CD-ROM. The MediaReactor installation program should auto-run. If not, go to your CD-ROM drive and double-click the **SetupDTMRX.exe** program.
3. You will be prompted to move through each step of the install by pressing the “next” or “finish” button. Carefully read the instructions that are provided.

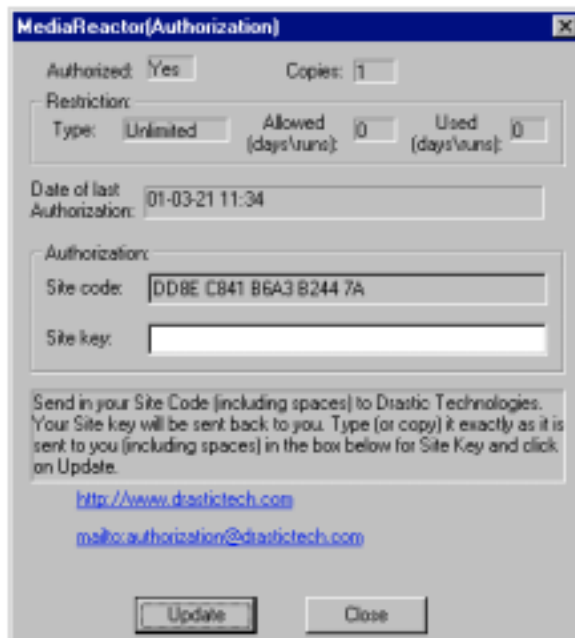
After all these files are installed, the machine may need to be restarted. The installation program will prompt you to restart.

Before you can use the MediaReactor SDK's ActiveX control it will have to be authorized. If your computer had to be restarted then the LicenseCheck utility will run when automatically. If not go to the directory you installed MediaReactor SDK to and run LicenseCheck.exe.

The authorization procedure involves providing Drastic Technologies with a site code after which you will receive the site key you need to authorize your license.

2.2.2 License Authorization

Selecting the **License Authorization** in the menu will open the **MediaReactor (Authorization)** window.



MediaReactor Authorization window

The top of the window will show your current authorization status. The Authorization section of the window is where you will obtain the **Site Code** and input your **Site Key**. Due to the length of the site code, we recommend you select it and use the copy function (Ctrl-C) to put it in the clipboard. It is absolutely imperative that this site code is correctly sent to Drastic or a site key will not be properly generated. Therefore, paste the site code into a message and email it to Drastic Technologies at authorization@drastictech.com.

You will receive a reply from us containing your site key. The site key is a string of numbers and letters generated by our copy protection software. This sequence must be entered into the site key field exactly as generated in order to authorize your license. It is for this reason that we recommend that you select the whole site key, and Copy and Paste it into the proper field.

We will attempt to respond as quickly as possible but be sure to give yourself a good time margin in requesting your authorization. Drastic Technologies' offices are located in the Eastern Standard Time Zone of North America and we will only be able to provide you with a code during regular business hours (Monday to Friday, 9am to 5pm EST).

Once you have received the reply with the site key, select it and copy it to the clipboard (Ctrl-C). Open MediaReactor and go to the **Authorization** section of MediaReactor (Authorization) window. Paste the site key into its box and push the **Update** button. This will authorize your license and allow you to run the MediaReactor ActiveX control.

2.2.3 Software License Issues

You should be aware of the following technical issues that will cause your **License** to fail:

Disk Failure	After replacing the drive and re-installing the software, contact Drastic to obtain a new authorization. Please explain the situation in your request.
Clock Change	Changing the clock will be detected and cause your license to fail. Confirm that the system clock is set correctly before authorizing MediaReactor.
Disk compression or de-fragmentation software.	Using a program like Norton Speed Disk can damage your license. If you wish to use one of these programs make sure you configure them not to move any *.ent, *.rst and *.key files.
Attempting to re-install	Deleting the current MediaReactor folder and re-installing the software to another location will void your license. This may also result in damaging your license in such a way as to require a complete reinstall of your whole system before you can install a retail version of MediaReactor.
Installing in different directories	You must install all the demo versions in the same directory. If you attempt to install them into another folder it will void your trial license.

2.2.4 Sample Projects

Included with the SDK are a set of example projects done in both Visual C++ with MFC and Visual Basic. They are in the file Samples.zip and can be decompressed to their respective directories.

2.2.5 Third Party Software Installation

There is software made by other companies that must be on your system for the MediaReactor software to function properly. Currently MediaReactor requires that Apple's QuickTime software must be installed.

3 Quick Start

In order to help developers get working with the MediaReactor SDK as quickly as possible a number of sample applications are provided.

3.1 Examples

3.1.1 Visual Basic – DTMRX_StepOne

This is the most basic translation application possible. Once the input and output files are set, the DisplayOutputDialog member is called to setup the output type. Once the output type is set, the translate button starts the translation which is monitored by the Progress member in the timer.

3.1.2 Visual Basic – DTMRX_StepTwo

This example uses the PreviewXXX members to open a media file and display its information.

3.1.3 Visual Basic – DTMRXTest

This is an example of a user interface that does setup and validation of parameters for a translation. The DisplayOutputDialog and SetOutputPreset are also available.

3.1.4 Visual C - DTMXMFC

This example uses Microsoft Visual C and MFC to create a translation interface.

4 Reference Guide

This guide provides information about functions, file types and formats that MediaReactor supports.

4.1 Functions

The supplied MediaReactor functionality is split into two independent sections:

1. PreviewXXX functions for getting source file information and generating preview images and sound.
2. Translation functions for enumerating available types, setting source and target files and initiating the translation.

The preview functions are found in the '*Function Group: File Info/Preview*' section. All other '*Function Group:*' sections refer to file translation.

4.1.1 Function Group: **Utility**

HRESULT CheckSystem([in] long hWnd, [in] BOOL fDisplayErrors)

Check the operating system for common installation errors. Display fill cause message boxes to be used for each error

4.1.2 Function Group: **File Info/Preview**

HRESULT PreviewOpen([in] BSTR FileName, [in] long VidHorizontalSize, [in] long VidVerticalSize, [in] long AudBitsPerSample, [in] long AudSampleRate, [in] long AudChannels)

PreviewOpen to get source file info and read uncompressed frames or samples

HRESULT PreviewReadVideo([in] long Frame, [out] long * pRGBABuffer, [in,out] long * BufferSize)

PreviewReadVideo returns resized video frames as RGBA data

HRESULT PreviewReadAudio([in] long StartSample, [out] long * pBuffer, [in,out] long * BufferLength)

PreviewReadAudio returns converted audio samples as PCM data

HRESULT PreviewClose()

PreviewClose closes the current preview file

HRESULT PreviewFileHorizontal([out, retval] long *pVal)

PreviewFileHorizontal returns the source files original number of horizontal pixels

HRESULT PreviewFileVertical([out, retval] long *pVal)

PreviewFileVertical returns the source files original number of vertical pixels

HRESULT PreviewFileVideoChannels([out, retval] long *pVal)

PreviewFileVideoChannels returns the source files video channels as a bit array

HRESULT PreviewFileBitDepth([out, retval] long *pVal)

PreviewFileBitDepth returns the number of bits per pixel in the source file's video

HRESULT PreviewFileAudioChannels([out, retval] long *pVal)

PreviewFileAudioChannels returns the source files audio channels as a bit array

HRESULT PreviewFileInfoChannels([out, retval] long *pVal)

PreviewFileInfoChannels returns the source files information channels as a bit array

HRESULT PreviewFileCodec([out, retval] long *pVal)

PreviewFileCodec return the codec used by the source file

HRESULT PreviewFileFrameRate([out, retval] double *pVal)

PreviewFileFrameRate returns the frame rate of the source file

HRESULT PreviewFileFrames([out, retval] long *pVal)

PreviewFileFrames returns the number of frames available in the source file

HRESULT PreviewFileSampleRate([out, retval] long *pVal)

PreviewFileSampleRate returns the samples rate of the source file's audio

HRESULT PreviewFileSamples([out, retval] long *pVal)

PreviewFileSamples returns the number of samples in the source file

HRESULT PreviewFileBitsPerSample([out, retval] long *pVal)

PreviewFileBitsPerSample returns the number of bits per sample in the source file's audio

4.1.3 Function Group: **Get Available Functions**

HRESULT CanTranslate([out, retval] BOOL *pVal)

If true, the control can translate from one file type to another

HRESULT CanPreviewRead([out, retval] BOOL *pVal)

If true, the control can read pcm audio and RGBA frames via the preview interface

HRESULT CanPreviewWrite([out, retval] BOOL *pVal)

If true, the control can write pcm audio and RGBA frames via the preview interface

HRESULT infoSupportedFileTypes([in] long FileCount, [out] BSTR * FileExt, [out] BSTR * ShortDesc, [out] BSTR * LongDesc, [out] long * pFlags, [out] long * pFileID)

Retrieve the supported output file types. FileIDs start at 0 and increment until an error return indicates the last type has been returned

HRESULT infoSupportedCodecTypes([in] long FileID, [in] long CodecCount, [out] BSTR * ShortDesc, [out] BSTR * LongDesc, [out] long * pFlags, [out] long * pCodecID)

Retrieve the supported output codec types. FileIDs start at 0 and increment until an error return indicates the last type has been returned

HRESULT infoSupportedProcessTypes([in] long ProcessCount, [out] BSTR * ShortDesc, [out] BSTR * LongDesc, [out] long * pFlags, [out] long * pProcessID)

Retrieve the supported processing types. FileIDs start at 0 and increment until an error return indicates the last type has been returned

4.1.4 Function Group: **Set File Output**

HRESULT DisplayOutputDialog([in] long hwndParent)

DisplayOutputDialog displays a dialog, which allows the user to setup the output parameters for the translation

HRESULT SetOutputPreset([in] long lTargetFileType, [in] BOOL fPAL25Fps)

Set the output to one of the standard supported types

4.1.5 Function Group: **Set Video Output**

HRESULT VOutFileType([in] long pVal)
HRESULT VOutFileType([out, retval] long *pVal)

Get/Set the main output file type using types returned by infoSupportedFileTypes

HRESULT VOutCodecType([in] long pVal)
HRESULT VOutCodecType([out, retval] long *pVal)

Get/Set the main output codec type using types returned by infoSupportedCodecTypes

HRESULT VOutHorizontalXPixels([in] long pVal)
HRESULT VOutHorizontalXPixels([out, retval] long *pVal)

Video Output Horizontal Pixels

HRESULT VOutVerticalYPixels([in] long pVal)
HRESULT VOutVerticalYPixels([out, retval] long *pVal)

Video Output Vertical Pixels

HRESULT VOutBitDepth([in] long pVal)
HRESULT VOutBitDepth([out, retval] long *pVal)

Video Output Bit Depth

HRESULT VOutQuality([in] long pVal)
HRESULT VOutQuality([out, retval] long *pVal)

Video Output Quality

HRESULT VOutDataRate([in] long pVal)
HRESULT VOutDataRate([out, retval] long *pVal)

Video Output Data Rate

HRESULT VOutKeyFrame([in] long pVal)
HRESULT VOutKeyFrame([out, retval] long *pVal)

Video Output Key Frame

4.1.6 Function Group: Set Audio Output

HRESULT AOutFileType([in] long pVal)
HRESULT AOutFileType([out, retval] long *pVal)

Get/Set the separate audio output file type using types returned by infoSupportedFileTypes

HRESULT AOutCodecType([in] long pVal)
HRESULT AOutCodecType([out, retval] long *pVal)

Get/Set the audio output codec type using types returned by infoSupportedCodecTypes

HRESULT AOutFrequency([in] long pVal)
HRESULT AOutFrequency([out, retval] long *pVal)

Audio Output Frequency

HRESULT AOutSampleSize([in] long pVal)
HRESULT AOutSampleSize([out, retval] long *pVal)

Audio Output Sample Size

HRESULT AOutChannels([in] long pVal)
HRESULT AOutChannels([out, retval] long *pVal)

Audio Output Channels

HRESULT AOutQuality([in] long pVal)
HRESULT AOutQuality([out, retval] long *pVal)

Audio Output Quality Setting

HRESULT AOutSeparateAudio([in] BOOL newVal)
HRESULT AOutSeparateAudio([out, retval] BOOL *pVal)

Set/Get if a separate audio file is going to be generated

HRESULT AOutDualMono([in] BOOL newVal)
HRESULT AOutDualMono([out, retval] BOOL *pVal)

Get/Set the separate audio output for mono or stereo output"

4.1.7 Function Group: **Set Input And Output Files**

HRESULT AddInput([in] BSTR FileName, [in] long StartFrame, [in] long EndFrame, [in] long AuxFrame, [in] long Flags)

Add an input file to be translated. This may be an audio video stream or separate audio and video files added individually using the flags variable

HRESULT AddOutput([in] BSTR FileName, [in] long Flags)

Add an output file to write to. Output of interleaved stream is normal, but separate audio and video streams may be created using the flags variable

4.1.8 Function Group: **Process**

HRESULT ProcessEnable([in] BOOL newVal)
HRESULT ProcessEnable([out, retval] BOOL *pVal)

Get/Set if the current process type is enabled

HRESULT ProcessType([in] long newVal)
HRESULT ProcessType([out, retval] long *pVal)

Get/Set the current process type (one of the values returned by infoSupportedProcessTypes)

HRESULT ProcessP1([in] long newVal)
HRESULT ProcessP1([out, retval] long *pVal)
HRESULT ProcessP2([in] long newVal)
HRESULT ProcessP2([out, retval] long *pVal)
HRESULT ProcessP3([in] long newVal)
HRESULT ProcessP3([out, retval] long *pVal)

Get/Set the P1 process property (threshold), P2 process property (edge detection) or P3 process property

4.1.9 Function Group: Translation

HRESULT NewTranslation()

Initialize for a new translation. If a translation is already running, this call will fail

HRESULT Translate()

Start the actual translation. Use the Progress property to monitor translation process

HRESULT TranslationActive([out, retval] BOOL *pVal)

Returns TRUE if a translation is currently active

HRESULT Progress([out, retval] long *pVal)

Current file translation progress (0 to 1000)

4.1.10 Function Group: **Advanced**

HRESULT AdvancedSetFileOption([in] long IOID, [in] long Option, [in,out] long * Setting)

AdvancedSetFileOption is not normally required. Sets odd file settings for particular file types

HRESULT AdvancedSetCodecOption([in] long FileID, [in] long CodecID, [in] long OptionID, [in,out] long *Setting)

AdvancedSetCodecOption is not normally required. Sets odd codec settings for particular codec types

4.2 Media Formats

MediaReactor supports the reading and writing of digital media files containing video, audio and still image data. The following section lists the supported file types and codecs that are available for use with them.

4.2.1 Video for Windows (AVI)

MediaReactor supports Microsoft's Video for Windows file type. It provides a set of custom codecs that support specific hardware standards. It also supports third party software codecs.

The custom compression/decompression modules appear with the other codecs that are installed in the system but they are built into the MediaReactor software. These codecs provide the access to the hardware-specific translation capabilities of MediaReactor.

Codec	Notes
Drastic VVW 3x00 series Native Format	3000 series Drastic Technologies Digital Disk Recorder
Matrox – DigiSuite Open DML Motion JPEG	Matrox DigiSuite and DigiSuite LE hardware. These platforms are used in systems running IMC – Incite, Discreet – edit*, in-sync- Speed razor and Adobe Premiere RT
Matrox – Rainbow Runner MJPEG	Matrox 's Rainbow Runner. This is a cost effective Motion JPEG solution from Matrox.
Matrox – M-JPEG Lossless	
Pinnacle / Miro – DC30 /	Pinnacle's Miro DC30 and DC50. These

Codec	Notes
DC50	platforms are used in systems running Adobe Premiere and After Effects.
Pinnacle Reeltime	Pinnacle Reeltime hardware. This platform supports Adobe Premiere and in-sync Speed Razor
Targa RTX Motion JPEG – Edit Option 2	Pinnacle TARGA 2000. These platforms are used in systems running Discreet – edit* , in-sync Speed Razor and Adobe Premiere RT
Uncompressed (no codec)	RGB uncompressed video

Video for Windows uses software codecs. These modules may come from a number of manufacturers. For maximum compatibility, Media Reactor will attempt to use any software codecs that are installed on the system.

MediaReactor will also install some software codecs to provide compatibility with hardware that is not in your system. These codecs will be available for any other application that can use Video for Windows.

The following is a list of codecs that MediaReactor will install:

Codec	Notes
Matrox DV / DVCAM	Matrox Digiserver DTV.
Matrox DVCPRO	Matrox Digiserver DTV.
Matrox DVCPRO50	Matrox Digiserver DTV.
Matrox MPEG 2 I-frame	Matrox Digiserver DTV.

The following is a list of third party codecs that have been tested with MediaReactor.

Codec	Notes
Cinepak	Cinepak codec from Radius
DivX - DIV3	
DivX - DIV4	
Intel 263	
Indeo® video 5.0	
Intel Indeo® Video R3.2	
Intel Indeo® Video R4.5	
Microsoft Video 1	(CRAM)
MS-MPEG4 V1	
MS-MPEG4 V2	
MS-MPEG4 V3	This codec does not encode files.
MS-RLE	

4.2.2 Open Media Framework Interchange (OMF)

Media Reactor supports the OMFI (Open Media Framework Interchange) file type. This is used by a number of products from Avid Technology Inc. MediaReactor has software codecs that

support Avid's NuVista, ABVB and Meridien compression hardware. MediaReactor provides software codecs to support the Targa compression hardware used in the MCXpress NT systems. MediaReactor also supports uncompressed YUV 4:2:2, TIFF RGB and RGBA video in the OMFI container.

MediaReactor is capable of working with either OMF Interchange Specification Version 1.0 or 2.1 files. All the supported video formats are available in both of these versions.

Codec	Notes
AVR 1	NuVista 320 X 240 (288PAL) 1 Field
AVR 12	ABVB 720 X 243 (288PAL) 2 Field
AVR 1e	NuVista 320 X 240 (288PAL) 1 Field
AVR 2	NuVista 320 X 240 (288PAL) 1 Field
AVR 25	NuVista 640 X 240 (288PAL) 2 Field
AVR 26	NuVista 640 X 240 (288PAL) 2 Field
AVR 27	NuVista 640 X 240 (288PAL) 2 Field
AVR 2e	NuVista 320 X 240 (288PAL) 1 Field
AVR 2m	ABVB 352 X 243 (288PAL) 1 Field
AVR 2s	ABVB 720 X 243 (288PAL) 1 Field
AVR 3	NuVista 640 X 240 (288PAL) 1 Field
AVR 3e	NuVista 320 X 240 (288PAL) 1 Field
AVR 3m	ABVB 352 X 243 (288PAL) 1 Field
AVR 3s	ABVB 720 X 243 (288PAL) 1 Field
AVR 4	NuVista 640 X 240 (288PAL) 1 Field
AVR 4e	NuVista 640 X 240 (288PAL) 1 Field
AVR 4m	ABVB 352 X 243 (288PAL) 1 Field
AVR 4s	ABVB 720 X 243 (288PAL) 1 Field
AVR 5	NuVista 640 X 240 (288PAL) 1 Field
AVR 5e	NuVista 640 X 240 (288PAL) 1 Field
AVR 5m	ABVB 352 X 243 (288PAL) 1 Field
AVR 6e	NuVista 640 X 240 (288PAL) 1 Field
AVR 6m	ABVB 352 X 243 (288PAL) 1 Field
AVR 6s	ABVB 720 X 243 (288PAL) 1 Field
AVR 70	ABVB 720 X 243 (288PAL) 2 Field
AVR 71	ABVB 720 X 243 (288PAL) 2 Field
AVR 75	ABVB 720 X 243 (288PAL) 2 Field
AVR 77	ABVB 720 X 243 (288PAL) 2 Field
AVR 8s	ABVB 720 X 243 (288PAL) 1 Field
AVR 9s	ABVB 720 X 243 (288PAL) 1 Field
JFIF 15:1s	Meridien 352 X 243 (288PAL) 1 Field
JFIF 2:1	Meridien 720 X 243 (288PAL) 2 Field
JFIF 2:1s	Meridien 352 X 243 (288PAL) 1 Field
JFIF 20:1	Meridien 720 X 243 (288PAL) 2 Field
JFIF 3:1	Meridien 720 X 243 (288PAL) 2 Field
JFIF 4:1s	Meridien 352 X 243 (288PAL) 1 Field
Uncomp422	Uncompressed YUV 720 X 243 (288PAL) 2 Field
TIFF RGB	Uncompressed RGB 720 X 243 (288PAL) 2 Field

Codec	Notes
Uncompressed	
Uncompressed RGBA	Uncompressed RGB with Alpha 720 X 243 (288PAL) 2 Field
MCXpress NT	Targa 720 X 486 (288PAL) 2 Field

4.2.3 QuickTime (MOV)

MediaReactor supports Apple's QuickTime file type. It provides a set of custom codecs that support specific software and hardware standards. It also supports Third Party software codecs.

The custom compression/decompression modules appear with the other codecs that are installed in the system but they are built into the MediaReactor software. These codecs provide the access to the hardware- or software-specific translation capabilities of MediaReactor.

Codec	Notes
Final Cut DV NTSC	Apple's Final Cut software.
Final Cut DV PAL	Apple's Final Cut software.
Media 100 Macintosh	
Media 100 and Finish NT	
Targa RTX Motion JPEG	
Motion JPEG A	Apple's Motion JPEG A
Motion JPEG B	Apple's Motion JPEG B
Avid AVR 1	NuVista 320 X 240 (288PAL) 1 Field
Avid AVR 12	ABVB 720 X 243 (288PAL) 2 Field
Avid AVR 1e	NuVista 320 X 240 (288PAL) 1 Field
Avid AVR 2	NuVista 320 X 240 (288PAL) 1 Field
Avid AVR 25	NuVista 640 X 240 (288PAL) 2 Field
Avid AVR 26	NuVista 640 X 240 (288PAL) 2 Field
Avid AVR 27	NuVista 640 X 240 (288PAL) 2 Field
Avid AVR 2e	NuVista 320 X 240 (288PAL) 1 Field
Avid AVR 2m	ABVB 352 X 243 (288PAL) 1 Field
Avid AVR 2s	ABVB 720 X 243 (288PAL) 1 Field
Avid AVR 3	NuVista 640 X 240 (288PAL) 1 Field
Avid AVR 3e	NuVista 320 X 240 (288PAL) 1 Field
Avid AVR 3m	ABVB 352 X 243 (288PAL) 1 Field
Avid AVR 3s	ABVB 720 X 243 (288PAL) 1 Field
Avid AVR 4	NuVista 640 X 240 (288PAL) 1 Field
Avid AVR 4e	NuVista 640 X 240 (288PAL) 1 Field
Avid AVR 4m	ABVB 352 X 243 (288PAL) 1 Field
Avid AVR 4s	ABVB 720 X 243 (288PAL) 1 Field
Avid AVR 5	NuVista 640 X 240 (288PAL) 1 Field
Avid AVR 5e	NuVista 640 X 240 (288PAL) 1 Field
Avid AVR 5m	ABVB 352 X 243 (288PAL) 1 Field
Avid AVR 6e	NuVista 640 X 240 (288PAL) 1 Field
Avid AVR 6m	ABVB 352 X 243 (288PAL) 1 Field

Codec	Notes
Avid AVR 6s	ABVB 720 X 243 (288PAL) 1 Field
Avid AVR 70	ABVB 720 X 243 (288PAL) 2 Field
Avid AVR 71	ABVB 720 X 243 (288PAL) 2 Field
Avid AVR 75	ABVB 720 X 243 (288PAL) 2 Field
Avid AVR 77	ABVB 720 X 243 (288PAL) 2 Field
Avid AVR 8s	ABVB 720 X 243 (288PAL) 1 Field
Avid AVR 9s	ABVB 720 X 243 (288PAL) 1 Field
Avid JFIF 15:1s	Meridien 352 X 243 (288PAL) 1 Field
Avid JFIF 2:1	Meridien 720 X 243 (288PAL) 2 Field
Avid JFIF 2:1s	Meridien 352 X 243 (288PAL) 1 Field
Avid JFIF 20:1	Meridien 720 X 243 (288PAL) 2 Field
Avid JFIF 3:1	Meridien 720 X 243 (288PAL) 2 Field
Avid JFIF 4:1s	Meridien 352 X 243 (288PAL) 1 Field

QuickTime uses software codecs. These modules may come from a number of manufacturers. For maximum compatibility, MediaReactor will attempt to use any software codecs that are installed on the system.

MediaReactor will also install some software codecs to provide compatibility with hardware that is not in your system. These codecs will be available for any other application that can use QuickTime.

The following is a list of codecs that MediaReactor will install.

Codec	Notes
10 Bit Y'CrCb	Viewgraphics VideoPump
8 Bit Y'CrCb	Viewgraphics VideoPump

The following is a list of third party codecs that have been tested with MediaReactor.

Codec	Notes
Apple Animation	
Apple BMP	
Apple Cinepak	
Apple Component Video – YUV 422	
Apple DV NTSC	
Apple DV PAL	
Apple Graphics	
Apple H 261	
Apple H 263	
Apple None	
Apple Photo JPEG	
Apple Planar RGB	
Apple PNG	
Apple TGA	
Apple TIFF	

Codec	Notes
Apple Video	
Intel Indeo Video 4.4	
Sorenson Video™ Compressor	

4.2.4 Audio Files

MediaReactor supports a number of audio file types. It currently works with PCM audio.

MediaReactor does not currently support any compressed audio formats.

File Type	Notes
8svx – Amiga Audio	
aiff – Apple – SGI Audio	
au – Sun Audio	
hcom – Apple Audio (old)	
maud – M Audio Format	
sf – Audio Format	
smp – Sample Vision Audio	
snd – NeXT Sound	
voc – Creative Labs	
Wav – Window Wave	

4.2.5 Still Image Files

MediaReactor supports a number of still image file types.

File Type	Notes
avs - X Still Image	
bmp - MS Windows Bitmap	
dcx - ZSoft Still Image	Paintbrush
dib - Device Independent Bitmap	MS Windows
icb - TGA Still Image Variant	Truevision Image Capture Board
inp – Chyron Infinite RGBA	
jpg - JPEG Still Image	Joint Photographic Experts Group format
mng - Multi Image Graphic	
pbm - Portable Bitmap	
pcd - Photo CD Format	
pcx - Paintbrush Image	Zsoft
pgm - Portable Graymap Format	
pic – Pict Still Image File	
png – Portable Network Graphic	
pnm – Portable Anymap Still Image	
ppm - Portable Pixmap Still Format	

File Type	Notes
psd – PhotoShop File	Adobe PhotoShop
ras - Sun Microsystems Raster	
rgb Raw RGB image file	
sgi – SGI Still Image	
sun - SUN Raster image	
tga – Targa Still	Truevision raster graphic
tiff - Tagged Image Format	
vda - TGA Image variant	
vst - Targa Vista Image Format	
xbm - X Windows bitmap (b&w)	
xpm - X Windows bitmap (color)	
yuv - Raw YUV Image	