Audio Xtra® Help

Installation

Getting Started

Xtra Function Reference

Error codes

Using the Xtra in Shockwave

File Type Extensions

Creating Projectors

Sound List Table

Environment Properties Table

How to Order & Register

Licensing & Availability

Technical Support

For up-to-date information please visit our web site:
xtras.tabuleiro.com

AUDIO XTRA HELP: INSTALLATION

The installation procedure is slightly different depending on the version of Director and platform used. Make sure you have administrative rights to create files in the directory where Director is installed on your system.

WINDOWS	MACINTOSH
Director 11/11.5	Director 11/11.5
Director MX 2004	Director MX 2004
Director MX	Director MX
Director 8.5	Director 8.5

AUDIO XTRA HELP: INSTALLATION: WINDOWS - DIRECTOR 11 / 11.5

INSTALLING THE XTRA ON WINDOWS - Director 11 / 11.5

Decompress the installation .zip file. This will unpack the Xtra, documentation and sample files to a folder named "Audio Xtra" on your machine. To install the Xtra, just copy the file Windows\AudioXtra.x32 to the Director 11 XTRAS folder. If your copy of Director 11 is installed at the default location, the Windows Xtra file will be located at:

C:\Program Files\Adobe\Adobe Director 11\Configuration\Xtras\AudioXtra.x32

Now you need to install the files necessary for creation of cross-platform projector for Mac OSX. Go back to the "Audio Xtra" directory where the Xtra files were unpacked. Open the **Mac Universal** directory. Now copy the file "Audio Xtra.cpio" to the "Configuration\Cross Platform Resources\Macintosh\Xtras" directory used by Director 11. In a default installation of Director this file will end up at the following location:

C:\Program Files\Adobe\Adobe Director 11\Configuration\Cross Platform Resources\Macintosh\Xtras\Audio Xtra.cpio

Finally, you need to edit the xtrainfo.txt file to include information about Audio Xtra. This information is used by the Shockwave and cross-platform publishing features in Director 11, to locate the files needed when assembling the Mac OSX version of your projector. The xtrainfo.txt file is located by default at:

C:\Program Files\Adobe\Adobe Director 11\Configuration\xtrainfo.txt

Double click the file to open it in notepad, or alternatively edit with any other text editor. You need to add the following line to the end of the file:

```
[#namePPC:"Audio Xtra", #nameW32:"AudioXtra.x32",
#package:"http://download.tabuleiro.com/packages/Audio/61/AudioXtra"]
```

You may want to customize this line in the future, to instruct Director to download Audio Xtra Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the Using the Xtra in Shockwave section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.

AUDIO XTRA HELP: INSTALLATION: WINDOWS - DIRECTOR MX 2004

INSTALLING THE XTRA ON WINDOWS - Director MX 2004

If you have not done so, we recommend updating to Director MX 2004 version 10.1 before installing the Xtra. This will allow creation of projectors for Mac Classic and OSX (Director MX 2004 without the update can only create cross platform projectors for OSX.)

Decompress the installation .zip file. This will unpack the Xtra, documentation and sample files to a folder named "Audio Xtra" on your machine. To install the Xtra, just copy the file Windows\AudioXtra.x32 to the Director MX 2004 XTRAS folder. If your copy of Director MX 2004 is installed at the default location, the Windows Xtra file will be located at:

C:\Program Files\Macromedia\Director MX 2004\Configuration\Xtras\AudioXtra.x32

Now you need to install the files necessary for creation of cross-platform projector for Mac OSX. Go back to the "Audio Xtra" directory where the Xtra files were unpacked. Open the **Mac Carbon** directory. Now copy the files "Audio Xtra.data" and "Audio Xtra.rsrc" files to the "Configuration\Cross Platform Resources\Macintosh\Xtras" directory used by Director MX 2004. In a default installation of Director these files will end up at the following locations:

C:\Program Files\Macromedia\Director MX 2004\Configuration\Cross Platform Resources\Macintosh\Xtras\Audio Xtra.data

C:\Program Files\Macromedia\Director MX 2004\Configuration\Cross Platform Resources\Macintosh\Xtras\Audio Xtra.rsrc

If you are running Director MX 2004 10.1, you can also install the files necessary for creation of cross-platform projector for Mac Classic. Again, go back to the "Audio Xtra" directory where the Xtra files were unpacked. Open the **Mac Classic** directory. Now copy the files "Audio Xtra.data" and "Audio Xtra.rsrc" files to the "Configuration\Cross Platform Resources\Classic\Xtras" directory used by Director MX 2004. In a default installation of Director these files will end up at the following locations:

C:\Program Files\Macromedia\Director MX 2004\Configuration\Cross Platform Resources\Classic\Xtras\Audio Xtra.data

C:\Program Files\Macromedia\Director MX 2004\Configuration\Cross Platform Resources\Classic\Xtras\Audio Xtra.rsrc

Finally, you need to edit the xtrainfo.txt file to include information about Audio Xtra. This information is used by the Shockwave and cross-platform publishing features in Director MX 2004, to locate the files needed when assembling the OSX and Classic versions of your projector. The xtrainfo.txt file is located by default at:

C:\Program Files\Macromedia\Director MX 2004\Configuration\xtrainfo.txt

Double click the file to open it in notepad, or alternatively edit with any other text editor. You need to add the following line to the end of the file:

```
[#namePPC:"Audio Xtra", #nameW32:"AudioXtra.x32",
#package:"http://download.tabuleiro.com/packages/Audio/61/AudioXtra"]
```

You may want to customize this line in the future, to instruct Director to download Audio Xtra Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the Using the Xtra in Shockwave section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.

AUDIO XTRA HELP: INSTALLATION: WINDOWS - DIRECTOR MX AND 8.5

INSTALLING THE XTRA ON WINDOWS - Director MX and Director 8.5

Decompress the installation .zip file. This will unpack the Xtra, documentation and sample files to a folder named "Audio Xtra" on your machine. To install the Xtra, just copy the file Windows\AudioXtra.x32 to the Director 8.5 or Director MX XTRAS folder. If you have previously installed an older copy of the Xtra (previous versions were named as resaudio.x32) make sure to remove or replace it.

These are the default locations of the Xtras folder for each application:

Director 8.5- C:\Program Files\Macromedia\Director 8.5\Xtras

Director MX- C:\Program Files\Macromedia\Director MX\Xtras

Finally, you need to edit the xtrainfo.txt file to include information about Audio Xtra. This information is used by the Shockwave publishing features in Director. The xtrainfo.txt file is located by default at:

Director 8.5 - C:\Program Files\Macromedia\Director 8.5\xtrainfo.txt

Director MX - C:\Program Files\Macromedia\Director MX\xtrainfo.txt

Double click the file to open it in notepad, or alternatively edit with any other text editor. You need to add the following line to the end of the file:

```
[#namePPC:"Audio Xtra", #nameW32:"AudioXtra.x32",
#package:"http://download.tabuleiro.com/packages/Audio/61/AudioXtra"]
```

You may want to customize this line in the future, to instruct Director to download Audio Xtra Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the Using the Xtra in Shockwave section of the documentation. Restart Director for the changes to take effect. The

Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button (Director MX).

AUDIO XTRA HELP: INSTALLATION: MACINTOSH - DIRECTOR 11 / 11.5

INSTALLING THE XTRA ON MAC OSX - Director 11 / 11.5

Double-click the installation .dmg file. This will mount a disk named "Audio Xtra" on your desktop.

The first step is to copy the Universal binary version of the Xtra, which will be used in the authoring environment and also when creating Mac OSX projectors, for both Intel and PPC machines. This file is located in the install disk image, at:

Audio Xtra/Mac Universal/Audio Xtra.xtra

This file needs to be copied to the Director 11 Xtras folder. The final pathname for the Xtra in a default installation of Director 11 will be:

OSX Volume Name/Applications/Adobe Director 11/Configuration/Xtras/Audio Xtra.xtra

Windows projectors can also be created directly on Director 11 running on Mac OSX after installation of the Windows version of the Xtra. It is located on the install disk, at:

Audio Xtra/Windows/AudioXtra.x32

Copy this file to the Cross Platform resources directory in Director 11, so that it will be available at:

OSX Volume Name/Applications/Adobe Director 11/Configuration/Cross Platform Resources/Windows/Xtras/AudioXtra.x32

Finally, you need to edit the xtrainfo.txt file to include information about Audio Xtra. This information is used by the Shockwave and cross-platform publishing features in Director 11, to locate the files needed when assembling the Windows version of your projector. The xtrainfo.txt file is located by default at:

OSX Volume Name/Applications/Adobe Director 11/Configuration/xtrainfo.txt

Double click the file to open it in TextEdit, or alternatively edit with another text editor. Make sure to save the file in plain text format, though. You need to add the following line to the end of the file:

```
[#namePPC:"Audio Xtra", #nameW32:"AudioXtra.x32",
#package:"http://download.tabuleiro.com/packages/Audio/61/AudioXtra"]
```

You may want to customize this line in the future, to instruct Director to download Audio Xtra Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the Using the Xtra in Shockwave section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.

AUDIO XTRA HELP: INSTALLATION: MACINTOSH - DIRECTOR MX 2004

INSTALLING THE XTRA ON MAC OSX - Director MX 2004

Double-click the installation .dmg file. This will mount a disk named "Audio Xtra" on your desktop.

The first step is to copy the OSX version of the Xtra, which will be used in the authoring environment and also when creating OSX projectors. This file is located in the install disk image, at:

Audio Xtra/Mac Carbon/Audio Xtra

This file needs to be copied to the Director MX 2004 Xtras folder. The final pathname for the OSX Xtra in a default installation of Director MX will be:

OSX Volume Name/Applications/Macromedia Director MX 2004/Configuration/Xtras/Audio Xtra

Director MX 2004 running on Mac OSX can also be used to create Classic projectors, for Mac OS versions 8 and 9. In order to enable this feature you need to copy the Classic version of Audio Xtra to the correct location in your Director MX installation. First locate the Classic version of Audio Xtra in the install disk:

Audio Xtra/Mac Classic/Audio Xtra

This file needs to be copied to the following location in the Director MX 2004 folder, to be used for cross-platform publishing. Copy it to:

OSX Volume Name/Applications/Macromedia Director MX 2004/Configuration/Cross Platform Resources/Classic MacOS/Xtras/Audio Xtra

Windows projectors can also be created directly on Director MX 2004 running on Mac OSX after installation of the Windows version of the Xtra. It is located on the install disk, at:

Audio Xtra/Windows/AudioXtra.x32

Copy this file to the Cross Platform resources directory in Director MX 2004, so that it will be available at:

OSX Volume Name/Applications/Macromedia Director MX 2004/Configuration/Cross Platform Resources/Windows/Xtras/AudioXtra.x32

Finally, you need to edit the xtrainfo.txt file to include information about Audio Xtra. This information is used by the Shockwave and cross-platform publishing features in Director MX 2004, to locate the files needed when assembling the Classic MacOS and Windows versions of your projector. The xtrainfo.txt file is located by default at:

OSX Volume Name/Applications/Macromedia Director MX 2004/Configuration/xtrainfo.txt

Double click the file to open it in TextEdit, or alternatively edit with another text editor. Make sure to save the file in plain text format, though. You need to add the following line to the end of the file:

[#namePPC:"Audio Xtra", #nameW32:"AudioXtra.x32", #package:"http://download.tabuleiro.com/packages/Audio/61/AudioXtra"]

You may want to customize this line in the future, to instruct Director to download Audio Xtra Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the Using the Xtra in Shockwave section of the documentation. Restart Director for the changes to take effect. The

Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.

AUDIO XTRA HELP: INSTALLATION: MACINTOSH - DIRECTOR MX

INSTALLING THE XTRA ON MAC OSX - Director MX

Double-click the installation .dmg file. This will mount a disk named "Audio Xtra" on your desktop.

The first step is to copy the OSX version of the Xtra, which will be used in the authoring environment and also when creating OSX projectors. This file is located in the install disk image, at:

Audio Xtra/Mac Carbon/Audio Xtra

This file needs to be copied to the Director MX Xtras folder. The final pathname for the OSX Xtra in a default installation of Director MX will be:

OSX Volume Name/Applications/Macromedia Director MX/Xtras/Audio Xtra

Director MX running on Mac OSX can also be used to create Classic projectors, for Mac OS versions 8 and 9. In order to enable this feature you need to copy the Classic version of Audio Xtra to the correct location in your Director MX installation. First locate the Classic version of Audio Xtra in the install disk:

Audio Xtra/Mac Classic/Audio Xtra

This file needs to be copied to the following location in the Director MX folder:

OSX Volume Name/Applications/Macromedia Director MX/Classic MacOS/Xtras/Audio Xtra

Finally, you need to edit the xtrainfo.txt file to include information about Audio Xtra. This information is used by the Shockwave and cross-platform publishing features in Director MX 2004, to locate the files needed when assembling the Classic MacOS version of your projector. The xtrainfo.txt file is located by default at:

OSX Volume Name/Applications/Macromedia Director MX/xtrainfo.txt

Double click the file to open it in TextEdit, or alternatively edit with another text editor. Make sure to save the file in plain text format, though. You need to add the following line to the end of the file:

```
[#namePPC:"Audio Xtra", #nameW32:"AudioXtra.x32",
#package:"http://download.tabuleiro.com/packages/Audio/61/AudioXtra"]
```

You may want to customize this line in the future, to instruct Director to download Audio Xtra Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the Using the Xtra in Shockwave section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window. The Xtra functions will also be listed when you click the message window Scripting Xtras button.

AUDIO XTRA HELP: INSTALLATION: MACINTOSH - DIRECTOR 8.5

INSTALLING THE XTRA ON MAC OS 8 AND 9 - Director 8.5

Running under OSX, double-click the installation .dmg file. This will mount a disk named "Audio Xtra" on your desktop. To install the Xtra just copy the file "Audio Xtra" from the Mac Classic folder to the Xtras folder of your Director 8.5 installation. The final pathname for the Xtra will be for example:

Macintosh HD:OS9 Applications:Macromedia Director 8.5:Xtras:Audio Xtra

Finally, you need to edit the xtrainfo.txt file to include information about Audio Xtra. This information is used by the Shockwave publishing features in Director. The xtrainfo.txt file is located in the directory where Director 8.5 was installed, for example at:

Macintosh HD:OS9 Applications:Macromedia Director 8.5:xtrainfo.txt

Double click the file to open it in SimpleText, or another editor capable of saving plain text files. You need to add the following line to the end of the file:

```
[#namePPC:"Audio Xtra", #nameW32:"AudioXtra.x32",
#package:"http://download.tabuleiro.com/packages/Audio/61/AudioXtra"]
```

You may want to customize this line in the future, to instruct Director to download Audio Xtra Shockwave packages from the same server that hosts your Shockwave applications. This is covered in more detail at the Using the Xtra in Shockwave section of the documentation. Restart Director for the changes to take effect. The Xtra should be listed when you issue the command "put the xtralist" in the message window.

AUDIO XTRA HELP: GETTING STARTED

Audio Xtra is a Scripting Xtra. Scripting Xtras are used to extend the Lingo language with new functions and datatypes. Unlike Asset Xtras there is no visual representation of a scripting Xtra in the Director interface, and you can not create castmembers or sprites. The creation of Scripting Xtras is done in your Lingo or JavaScript syntax scripts, by using the #new Lingo keyword.

The first step is to download and install the Audio Xtra, following the instructions in the installation page. Now that it's installed, let's verify that the installation was successful. You should see the Audio Xtra entry in the Scripting Xtras context menu, appearing at the top of the message window. Selecting the Audio Xtra submenu and the "put interface" entry will output a list of all commands understood by Audio Xtra in the message window. You can also use the following command

Lingo:

```
put the xtralist
```

JavaScript syntax:

```
trace(_player.xtraList)
```

to verify which Xtras are installed, including the version number for each one.

THE SOUNDLIST

The soundlist is Audio Xtra's internal list of sounds that it is currently working with. The source of the sound can be a file, a cast member or a sound recorded into RAM, but the sound must be loaded into Audio Xtra's sound list before other Audio Xtra commands can work with it. There are 3 ways to add a sound to Audio Xtra's sound list:

1. Record a new sound, which automatically adds it to the sound list

2. Load an existing sound file or sound cast member into the list using axLoadSound or axLoadSoundIntoRAM

3. Save a sound already in the sound list to a different format using axConvertToMember or axConvertToFile.

Use axGetSoundList to determine which sounds are currently on the sound list.

THE SOUNDS DIRECTORY

Almost all commands in Shockwave require a sounds directory set by the user. The sounds directory is used as the path to record to and the path to load sounds from. You prompt the user for the path using axPromptForSoundsDirectory.

RECORDING A SOUND

You can record sound to an external file, a cast member, or straight to RAM. You must perform the following steps to record:

1. Set a sounds directory, or if you are going to record to a file in Shockwave, set the environment property useDswMediaDirectory to TRUE.

2. Issue axOpenRecorder and get no error back.

3. Issue either axRecordSoundToFile, axRecordSoundToRAM or axRecordSoundToMember.

4. Issue axStopRecording to stop recording. RAM or member based recordings will stop automatically when the buffer is full.

5. Issue axCloseRecorder to close the recorder.

PLAYING A SOUND

Once a sound is on the sound list you can play it using axPlay and stop it with axStop.

Examples:

```
axPlay("New Sound")

axStop("New Sound")
```

The Audio Xtra sample movie available in the TUTORIALS AND SAMPLES section at our website includes a full implementation of a simple application used to record, play and export audio files, complete with error checking and behaviors you can re-use on your files. The sample application was built to demonstrate best practices when accessing Audio Xtra functions and we strongly encourage you to re-use as much code from this sample as possible.

AUDIO XTRA HELP: XTRA FUNCTIONS

The following is a list of scripting functions available after installation of Audio Xtra.

axRegister
axOpenRecorder
axCloseRecorder
axRecordSoundToFile
axRecordSoundToRAM
axRecordSoundToMember
axServiceRecording
axStopRecording
axPauseRecording
axResumeRecording
axGetInputLevel
axLoadSound
axLoadSoundIntoRAM

axRemoveSound
axDeleteSound
axGetSoundList
axSetSoundInfo
axGetSoundInfo
axPlay
axStop
axGetStatus
axGetVersion
axSetEnvironmentInfo
axGetEnvironmentInfo
axGetFreeSpace
axGetEmptyMember

axConvertToFile
axConvertToMember
axConvertToSFData
axBytesToMilliseconds
axMillisecondsToBytes
axSetForegroundColor
axSetBackgroundColor
axPlotWaveform
axAskPermission
axPromptForSoundsDirectory
axCompressToOggFile
axCompressToOggData
axCompressToOggByteArray

axCompressToOggBase
axLoadOggSoundIntoR
axLoadOggByteArrayI
axLoadOggBase64Data

axRegister([1111,2222,3333]) - global function, used to register Audio Xtra. It can be called at any time, usually when the Director movie starts. Unregistered versions of the Xtra are fully functional for evaluation purposes, but they will display a warning the first time an Audio Xtra function is used. Returns TRUE if registration is successful, FALSE otherwise.

Audio Xtra serial number are strings, and have the generic format AUXZZ-1111-2222-3333, where ZZ is the major Xtra version. In order to protect your serial number from being included as a string in your Director projectors or dcr movies, the axRegister function requires only the three groups of numbers 1111, 2222 and 3333 inside a Director list. Leading zeroes do not need to be entered.

An example: if your serial number is AUX50-0123-4567-0089 then you should register using the following command, usually on a startmovie handler:

Lingo:

```
axRegister([123, 4567,89])
```

JavaScript syntax :

axRegister(list(123, 4567,89))

axOpenRecorder(bufferSizeInteger) where bufferSizeInteger is the size in bytes of the RAM buffer to reserve when recording to RAM. Returns an error number or 0 for no error. This command pens the sound recorder and allocates space in RAM for recording a sound to RAM or member. The bufferSize must be the maximum expected size of the recorded sound. The minimum recording buffer size for RAM recording is 10K on Windows and 20K on Mac. Passing a buffer size parameter of less than the required minimum will cause axOpenRecorder to fail and return an error.

For recording to a file, set bufferSize to 0 because a RAM buffer will not be used for recording. Setting a bufferSize other than 0 for recording to a file wastes RAM, since the memory will be allocated but never used. If you are recording to RAM or recording to a member keep in mind that after recording the sound will be copied out of the buffer to either a RAM location or a member, which means that the total memory requirement is actually twice the size of the sound. For instance if you set a buffer size of 40K and record a 40K sound into it you will need 80K of free memory to accomplish it - 40 for the buffer and 40 to store the finished sound in RAM or in a member.

The recorded sound will use the system default values for recording depth, sampling rate and number of channels, unless you have used axSetEnvironmentInfo to set other values. It is recommended to set the values for depth, rate and number of channels before the recorder is open. Settings changed while the recorder is open will be ignored unless the recorder is closed and reopened.

Example:

err = axOpenRecorder(102400)

-- sets the recording RAM buffer to 100K

If axOpenRecorder returns an error, you must use axCloseRecorder, before trying to open the recorder again, or recording may not function correctly.

axCloseRecorder() - Closes the sound recorder and releases the sound recording driver for use by other applications. The sound recorder must be closed before you can play sound if your sound card is not full duplex. Returns an error number or 0 for no error.

Example:

on mouseUp

axStopRecording()

axCloseRecorder()

end

axRecordSoundToFile(soundNameString, filePathString) where soundNameString is the name for the sound that will be added to the sound list and filePathString is the full file path or file name to save the recorded sound to in the sounds directory. Returns an error number or 0 for no error. An error will be returned if recording could not begin for some reason. In Shockwave the error -8718: "This feature is not supported in Shockwave" returns if the file already exists, for security reasons. axDeleteSound can be used in this case if the file has been created in the same recording section, or you must supply a different filename. If an error happened during recording, that will be returned from the axStopRecording command.

If a full path is not given, records sound to the specified file name in the sounds directory previously set with axSetEnvironmentInfo or axPromptForSoundsDirectory ,and adds the sound to the sound list. The recorder must be open for this command to work.

If a sounds directory is set and you pass a full path to the file for filePathString, the path is appended to the sounds directory path and will probably not result in a valid recording.

Recording in Shockwave will not begin until the user responds to any permissions dialog. If you are doing timed recording, start your timer after this command rather than before.

Although you can convert recorded sounds to other formats after recording, on Win you must record initially to a WAVE file and you must specify WAV as the file extension . On Mac you must record initially to an AIF file.

Note: Audio Xtra handles sound file extensions differently on the Mac and PC platform.

Examples:

-- next line contains no path so the sounds directory is assumed

err = axRecordSoundToFile ("New Sound", "TEST1.WAV")

err = axRecordSoundToFile ("Another", "C:\TEMP\TEST1.WAV")

axRecordSoundToRAM(soundNameString) where soundNameString is the name for the sound that will be added to the sound list. Returns an error number or 0 for no error. An error will be returned if recording could not begin for some reason. If an error happened during recording, that will be returned from the axStopRecording command. Records sound to memory and adds the sound to the sound list. The recorder must be open for this command to work.

Example:

err = axRecordSoundToRAM ("New Sound")

axRecordSoundToMember(soundNameString, memberReference) where soundNameString is the name for the sound that will be added to the sound list and memberReference is the member to hold the recorded sound. Returns an error number or 0 for no error. An error will be returned if recording could not begin for some reason. If an error happened during recording, that will be returned from the asStopRecording command. Records sound to to the specified member and adds the sound to the sound list. You can use the utility method axGetEmptyMember to find an empty cast member to receive the sound. The recorder must be open for this command to work.

Example:

err = axRecordSoundToMember ("Audio 1", axGetEmptyMember())

axServiceRecording() - Previous versions of Audio Xtra required this function to be called repeatedly during the recording process. In version 5.0 and later this is no longer necessary, as the Xtra will automatically service recording during idle events. However, this function is still present to provide compatibility with older content and Shockwave movies authored with previous versions.

axStopRecording() - Returns an error number or 0 for no error. Stops recording and creates a sound file or new member if the sound was designated to record to a file or to a member. A new sound in the process of recording will not appear in the sound list until axStopRecording() is called.

The amount of time you can record is determined by the size of the buffer set when you open the recorder. If you exceed the size of the buffer, recording will automatically stop. If that happens, and you then issue axStopRecording, Audio Xtra will return -8809, No sound is recording. A sound will be created with the recording up to the point the buffer filled up.

Example:

err = axStopRecording()

axPauseRecording() - Returns an error number or 0 for no error. Pauses recording. Does not create a new file or member because the sound is not yet finished. Waits for a resume to continue recording.

If the RAM buffer is too full to later resume recording, this command can return a -8820 error, and what has been recorded will be saved as a sound if there is enough RAM to do so.

Example:

err = axPauseRecording()

axResumeRecording() - Returns an error number or 0 for no error. Resumes a paused recording.

If the RAM buffer is too full to continue recording, this command can return a -8820 error, and what has been recorded will be saved as a sound if there is enough RAM to do so.

Example:

err = axResumeRecording()

axGetInputLevel() - Returns an integer between 0 and 255, or a negative error number. This command is only available on the Mac platform. Returns the current sound input volume level. The sound recorder must be open to get a level. On OSX, Director must be running to get a level. Calling the function from the Message Window while Director is stopped will always return 0.

Example:

axOpenRecorder(19249)

level = axGetInputLevel()

axLoadSound(soundNameString, typeString, memberRefOrFilePathString) where soundNameString is the name for the sound in the sound list, typeString is either "file" or "member" and memberRefOrFilePathString is either a member reference or a file path string. Returns an error number or 0 for no error.

Puts a sound that exists either in the Director movie's castlib or in an external file into Audio Xtra's sound list so that it can work with the member or file. If a file is specified, using this command creates a pointer from the sound list to the file but does not actually load the file into memory. If a member is specified however, using this command loads the member in to memory. It does the equivalent of Lingo's load member command.

If a sounds directory has been set with axSetEnvironmentInfo or axPromptForSoundsDirectory specifying a file name with no file path will automatically load the specified file from the sounds directory. You cannot specify a full file path in Shockwave. You must specify only a file name, which will load the sound file of that name from the sounds directory.

Note: Audio Xtra handles sound file extensions differently on the Mac and Windows platform.

Example:

axLoadSound("Happy Birthday","member",member 5)

axLoadSound("Narration1","file","My CD:sound:narr1.aif")

Import File Formats Supported

Format	Supported
AIFF	Yes
WAVE	Yes
AU	Yes
MP3	No
SWA	No

axLoadSoundIntoRAM(soundNameString, filePathString) where soundNameString is the name for the sound in the sound list and filePathString is the file path for the sound file to be loaded. Returns an error number or 0 for no error.

Puts an existing external sound file into Audio Xtra's sound list so that it can work with the file and loads the file into memory. This command is not available for cast members because axLoadSound automatically loads them in to memory. A sound loaded in to RAM will play back smoothly. You may want to use this function for a sound that skips when playing from disk.

If a sounds directory has been set with axSetEnvironmentInfo or axPromptForSoundsDirectory specifying a file name with no file path will automatically load the specified file from the sounds directory. You cannot specify a full file path in Shockwave. You must specify only a file name, which will load the sound file of that name from the sounds directory.

Note: Audio Xtra handles sound file extensions differently on the Mac and Windows platform.

Example:

axLoadSoundIntoRAM("Bird Song","D:\TEMP\BIRD.WAV")

-- loads BIRD.WAV from the sounds directory

axLoadSoundIntoRAM("Bird Song","BIRD.WAV")

axRemoveSound(soundNameString) where soundNameString is the name of a sound present in the sound list. Returns an error number or 0 for no error. Removes a sound from the sound list. If the sound was recorded into RAM, this command also releases the memory used by the sound.

Example:

axRemoveSound("Bird Song")

axDeleteSound(soundNameString) where soundNameString is the name of a sound present in the sound list. Returns an error number or 0 for no error. Removes a sound from the sound list and deletes its source media. The source member is deleted from the Director cast or the source file is deleted from the hard drive. If the sound was recorded into RAM, this command also releases the memory used by the sound. In Shockwave this command will only delete a sound file if it was recorded during the current session.

Note: The recorder must be closed (axCloseRecorder) or this command may return the error "-8224 file permission, device in use" and not delete the file, although the file will disappear from the sound list.

Example:

err = axDeleteSound("Narration 1")

axGetSoundList()

axGetSoundList(#all)

axGetSoundList(propertySymbol, propertySymbol...) where propertySymbol are the names of properties requested for listing. Returns an error number or a list containing the requested information for each sound on the sound list.

This command when issued with no arguments returns a list of the sound names in the sound list. When called with #all it returns the full sound list with every possible property and value listed for each sound on the list. When passed a variable number of property symbols, the listing still contains all of the sounds, but only returns the properties requested for each sound. See the Sound List Table for a list of the properties that can be requested with this command.

Example:

put axGetSoundList()

-- ["Song", "Horn Fanfare"]

put axGetSoundList(#all)

-- ["Song": [#name: "Song", #source: "file", #format: "WAVE", #start: 3991, #end: 0, #totalLength: 3998, #length: 3998, #position: 0, #depth: 8, #rate: 11127, #numChannels: 1, #volume: 255, #channel: 0, #preLoad: 1, #interleaveSeeking: 1], "Horn Fanfare": [#name: "Horn Fanfare", #source: "file", #format: "AIFF", #start: 4118, #end: 0, #totalLength: 4129, #length: 4129, #position: 0, #depth: 8, #rate: 11128, #numChannels: 1, #volume: 255, #channel: 0, #preLoad: 1, #interleaveSeeking: 1]]

put axGetSoundList(#source,#format)

-- ["Song": [#source: "file", #format: "WAVE"], "Horn Fanfare": [#source: "file", #format: "AIFF"]]

axSetSoundInfo(soundNameString, propertyString, propertyValue) where soundNameString is the name of a sound present in the sound list, propertyString is the name of the property to set and propertyValue is the new value for the property. Returns an error number or 0 for no error. Replaces an existing sound property value with the new value. See the Sound List Table for a list of the available properties and the subset that can be changed with this command.

Example:

axSetSoundInfo("Song","volume",125)

axGetSoundInfo(soundNameString, propertyString) where soundNameString is the name of a sound present in the sound list and propertyString is the name of the property to read. Returns the property value requested or a negative error number. Reads the requested property for the sound specified. See the Sound List Table for a list of the available properties. Since the sound list is formatted as a Lingo property list you can also read it directly using Lingo property list commands on the list returned from axGetSoundList()

Example:

```
put axGetSoundInfo("Song","format")

-- "WAVE"
```

axPlay(soundNameString) where soundNameString is the name of a sound present in the sound list. Returns an error number or 0 for no error. Plays the specified sound, which must be present in the sound list. On Windows, if the sound card is not full duplex you must use axCloseRecorder first or the sound will not play, or play with distortion.

Example:

```
err = axPlay("Song")
```

axStop(soundNameString) where soundNameString is the name of a sound present in the sound list. Returns an error number or 0 for no error. Stops playing the specified sound. Resets the position to 0.

Example:

```
err = axStop("Song")
```

axGetStatus(soundNameString) where soundNameString is the name of a sound present in the sound list. Returns a string describing the playback status of the sound or an error number if an error occurred. Returns the current playback status of a sound in the sound list.

The possible return values from a GetStatus call are:

"stopped" - Not playing

"playing" - Playing

"recording" - Recording

"record paused" - Paused while recording

"invalid" - Bad sound name argument passed

"error" - Not enough memory to complete the command - very unlikely

Example:

status = axGetStatus("Narration")

put status

-- "stopped"

axGetVersion() - Returns a string containing version number of Audio Xtra.

Example:

put axGetVersion()

-- "6.0"

axSetEnvironmentInfo(infoTypeString, newValue) where infoTypeString is the name of the environment property to set and newValue is the new value for the property. Returns an error number or 0 for no error. Sets a property that applies to Audio Xtra's recording and playback environment as a whole, rather than to a particular sound. The Environment Properties Table lists the available properties. Note that some properties only apply to one platform and some cannot be set.

Example:

sysTemp = axGetEnvironmentInfo("tempPath") axSetEnvironmentInfo ("soundsDirectory", sysTemp)

axGetEnvironmentInfo(infoTypeString) where infoTypeString is the name of the environment property to read. Returns a value for the specified property or a negative error number. Reads the value for a property that applies to Audio Xtra's recording and playback environment as a whole, rather than to a particular sound. The Environment Properties Table lists the available properties. Some properties only apply to one platform.

Example:

put axGetEnvironmentInfo("availableSoundFormats")

-- "AIFF

WAV"

axGetFreeSpace()

axGetFreeSpace(driveNameString) where driveNameString is the drive letter on Windows or the name of the volume on Mac. Returns a positive number representing number of free K (1024 bytes per K) or a negative error number. Returns the free space on the disk to help in estimating recording time available. Use axBytesToMilliseconds to calculate recording time available based on space. If no argument is passed and the sounds directory has already been specified, it returns the free space on the drive containing the sounds directory.

In Shockwave, you can only use this command with no arguments to get the space for the drive containing the sounds directory. Passing an arbitrary drive will return an error in those environments.

Note: If you make repeated calls to the function in a loop you may get a -8234 error return. The system calculates the free space in real time and it may take as long as a second for the system to clean up after the call. Make repeated calls to this function no more frequently than one second apart for best results.

Example:

freeRecordingK = axGetFreeSpace("Macintosh HD")

recordingSeconds = axBytesToMilliseconds(freeRecordingK * 1024) / 1000 axGetEmptyMember()

axGetEmptyMember(castlibNumInteger) where castlibNumInteger is an optional cast library number. Returns a full member reference for the empty member slot. Returns a member reference for the first empty cast member. If no castlib is specified, the internal castlib (1) is used. If a castlib number is specified, that castlib is used. Use this function to easily locate an empty member slot to pass to axRecordSoundToMember.

Example:

put axGetEmptyMember()

-- (member 55 of castLib 1)

put axGetEmptyMember(2)

-- (member 4 of castLib 2)

err = axRecordSoundToMember("new sound",axGetEmptyMember())

axConvertToFile(soundNameString, filePathString, formatString) where soundNameString is the name of a sound present in the sound list, filePathString is the file name or full file path to save new file to and formatString is one of the supported file formats for export. Returns an error number or 0 for no error. Saves an existing sound in the sound list to a file on the hard drive. The environment properties convertSampleRate, convertSampleDepth and convertNumChannels determine the properties of the converted file, and should be set using axSetEnvironmentInfo before this command is executed. Does not add the sound to the sound list. Use axLoadSound to add the new sound to the sound list.

The available save formats to choose from for the new sound depend on the Xtras loaded, since some of Director's MIX Xtras are used to save the file. See the Getting Started section for a list of all of the possible Xtras to include. axGetEnvironmentInfo("availableSoundFormats") will return the current formats provided by loaded Xtras. Additionally, there are some limitations, as shown in the Export File Formats Supported table below.

Export File Formats Supported

Format	String	Director 8.5, Director 11	Director MX, MX 2004	Extension
AIFF	"AIFF"	Yes	Yes	.AIF
WAVE	"WAVE"	Yes	Yes	.WAV
Sun AU	"AU"			.AU

		Yes, source file must be 8 bit depth	Yes, source file must be 8 bit depth	
MPEG 3	"MPEG3"	No	with significant restrictions *	.MP3
Shockwave Audio	"SWA"	No	with significant restrictions *	.SWA

* Please consult technote AU004 at the support area on our site for more information about restrictions in MP3 exporting from Audio Xtra.

If a sounds directory has been set with axSetEnvironmentInfo or axPromptForSoundsDirectory specifying just a filename with no path for filePathString will automatically create the specified file in the sounds directory. You cannot specify a full file path in Shockwave. You must specify only a file name, which will create the specified file in the sounds directory.

The filename, whether it is at the end of a path or by itself, must include a file extension that matches the specified file type on the PC. Refer to the File Format table above for the correct extension.

Note: Audio Xtra handles sound file extensions differently on the Mac and PC platform.

Limitations

When converting to AU files you must use settings of convertSampleRate = 8000, convertSampleDepth = 0, convertNumChannels = 0. The sound you are converting from must have an 8-bit sample depth. Any other settings will produce corrupted files. This problem lies in the Sun AU Import Export Xtra (version 8.0).

Example:

```
-- Saves existing sound "Recorded" in sound list to

-- the file "PRACTICE.WAV"

axConvertToFile ( "Recorded", "C:\MYDIR\PRACTICE.WAV", "WAVE")

-- Saves existing sound "BirdSong" in sound list to

-- the file "BIRD.AU" in the sounds directory.

axConvertToFile ( "BirdSong", "BIRD.AU", "AU")
```

axConvertToMember(soundNameString, newMemberNameString, newMemberReference) where soundNameString is the name of a sound present in the sound list, newMemberNameString is the name to use for the new cast member and newMemberReference is the member to put new sound into. The environment properties convertSampleRate, convertSampleDepth and convertNumChannels determine the properties of the converted file, and should be set using axSetEnvironmentInfo before this command is executed. Returns an error number or 0 for no error. Saves an existing sound in the sound list to an internal Director cast member. Does not add the sound to the sound list. Use axLoadSound or axLoadSoundIntoRAM to put the new sound onto the sound list. The format of the saved sound is Director's native internal sound format.

Example:

```
err = axConvertToMember("BirdSong","NewSong", axGetEmptyMember())
```

axConvertToSFData(soundNameString, filePathString, formatString) where soundNameString is the name of a sound present in the sound list, filePathString is the temporary file name created during the conversion and formatString is one of the supported file formats for export (see axConvertToFile for limitations.) Returns a property list compatible with ShockFiler's support for sending vList binary data files. The environment properties convertSampleRate, convertSampleDepth and convertNumChannels determine the properties of the converted file, and should be set using axSetEnvironmentInfo before this command is executed. This function was introduced in Audio Xtra 6, and provides an easier way for reducing the sample rate and size and preparing sound files to be uploaded with ShockFiler.

Example:

```
axSetEnvironmentInfo("useTempDirectory", 1)

axSetEnvironmentInfo("convertSampleRate", 11025)

axSetEnvironmentInfo("convertSampleDepth", 8)

axSetEnvironmentInfo("convertNumChannels", 1)



dataToSend = axConvertToSFData("recordedSound","tempfile.wav","WAVE")



remoteFileName = "recordedsound.wav"

remoteFileType = #bina
```

remotefileProperties = []

doappend = 0

showDialog = 1

dialogTitle = "Sending sound file ..."

error = sf_Send ("host", "directory", "usersname", "password" ,dataToSend, '', remoteFileName, remoteFileType, remotefileProperties, doappend, showDialog,dialogTitle)

axBytesToMilliseconds(numBytesInteger)

axBytesToMilliseconds(numBytesInteger, samplerateInteger, sampledepthInteger, numchannelsInteger) where numBytesInteger is the number of bytes and cannot exceed Lingo's the maxinteger, samplerateInteger is the sample rate to calculate, sampledepthIntegeris the sample depth to calculate and numchannelsInteger is the number of channels to calculate. Returns a positive number of milliseconds or a negative error number.Will not return a value higher than Lingo's the maxinteger. Converts a number of bytes to a sound duration in milliseconds.

If only one parameter is passed, Uses the current values set for recordSampleRate, recordSampleDepth and recordNumChannels as returned from axGetEnvironmentInfo to determine the disk space. If samplerate, depth and channels are passed, the calculation is done using those values and the recorder does not have to be open.

Example:

freeRecordingK = axGetFreeSpace("Macintosh HD")

recordingSeconds = axBytesToMilliseconds(freeRecordingK * 1024) / 1000

axMillisecondsToBytes(numMillsecsInteger)

axMillisecondsToBytes(numMillsecsInteger, samplerateInteger, sampledepthInteger, numchannelsInteger) where numMillsecsInteger is the number of milliseconds and cannot exceed Lingo's the maxinteger, samplerateInteger is the sample rate to calculate, sampledepthInteger is the sample depth to calculate and numchannelsInteger is the number of channels to calculate. Returns a positive number of bytes or a negative error number. Will not return a value higher than Lingo's the maxinteger. Converts a number of milliseconds of sound duration to an amount of disk space in bytes.

If only one parameter is passed, Uses the current values set for recordSampleRate, recordSampleDepth and recordNumChannels as returned from axGetEnvironmentInfo to determine the disk space. If samplerate, depth and channels are passed, the calculation is done using those values and the recorder does not have to be open.

Example:

diskSpace = axMillisecondsToBytes (9098777, 22050, 8, 2)

axSetForegroundColor(redInteger, greenInteger, blueInteger) where redInteger is the value for RGB red between 0 and 255, greenInteger is the value for RGB green between 0 and 255 and blueInteger is the value for RGB blue between 0 and 255. Returns an error number or 0 for no error. Sets the foreground color for the plotted waveform graphic. The default is white.

Example:

axSetForegroundColor(0,255,0)

axSetBackgroundColor(redInteger, greenInteger, blueInteger) where redInteger is the value for RGB red between 0 and 255, greenInteger is the value for RGB green between 0 and 255 and blueInteger is the value for RGB blue between 0 and 255. Returns an error number or 0 for no error. Sets the background color for the plotted waveform graphic. The default is black.

Example:

axSetBackgroundColor(125,0,255)

axPlotWaveform(soundNameString, displayMemberRef, widthInteger, heightInteger, resolutionSymbol, channelToPlotInteger) where soundNameString is the name of a sound present in the sound list; displayMemberRef is the member slot to create plot image and this can be an existing graphic member or an empty member; widthInteger is the width in pixels of plot image, timeBase param will override this; heightInteger is the height in pixels of plot image; resolutionSymbol is #low: 10-20 dots per X, #medium 30-50 dots per X, #high - 80-100 dots, #maximum - all dots; and channelToPlot is the number of the channel to plot. Returns an error number or 0 for no error.

Creates a plot graphic of the selected sound in the sound list and puts it into a bitmap cast member. The cast member specified can either be an existing bitmap cast member or an empty member slot. Audio Xtra cannot plot compressed sounds. Allocates memory the size of the environment variable plot buffer to perform the plot, and releases the memory after the plot. An out of memory error returned from this call means that a plot buffer of the specified size could not be allocated.

Note: very small sound files plotted over large-width plots may be not plottable at lower resolutions and will return an error. Try plotting at a higher resolution. This is a math limitation. Audio Xtra cannot create a plot image where there are more pixels horizontally than sound samples to plot.

Example:

```
axPlotWaveForm("BirdSong",member "plotImage",300,50,#medium,1)

-- Creates a plot bitmap in cast member "plotImage"

-- that is 300 pixels wide and 50 pixels high, at medium resolution.
```

axAskPermission(actionString, actionString ...) where actionString is the variable number of actions that user should be prompted for permission for. Returns 1 if the user granted permission, 0 if not, or a negative error number. Obtains permission from the user to perform a potentially unsafe action when the movie is running in Shockwave. You can pass one argument or multiple arguments to let the user grant multiple permissions at once. The following table shows the list of possible action strings and the corresponding Audio commands they ask permission for.

Audio Xtra commands requiring permission

actionString	Audio Xtra Function
inputLevels	axGetInputLevel
loadFiles	axLoadSound (if the source sound is a file), axLoadSoundIntoRAM, axConvertToMember (if the source sound is a file)
deleteFiles	axDeleteSound (if the source sound is a file)
recordFiles	axRecordToFile
recordMembers	axRecordToMember
freeSpace	axGetFreeSpace
plotWaveforms	axPlotWaveform
saveToFiles	axConvertToFile
saveToMembers	axConvertToMember (if the source sound is in RAM)

convertFiles axConvertToFile (if the source sound is a file)

The dialog put up by this command obtains permission for the requested function(s) for the duration of the session. See the Shockwave security section for an overview of working with Audio Xtra in that environment. To use this command in authoring for testing, you must set the environment property "simulateShockwave" to TRUE using axSetEnvironmentInfo.

Example:

```
userResponse = axAskPermission("deleteFiles","recordFiles")
```

axPromptForSoundsDirectory() - Returns 1 if the user selected a directory, 0 if the user cancelled the dialog, or a negative error number. Displays a folder selection dialog to allow the user to pick a sounds directory. If the user chooses a directory the environment property "soundsDirectory" will contain the path to the directory and "soundsDirectoryIsSet" will be TRUE. Use axGetEnvironmentInfo to query these properties.

Once a sounds directory is set, you need only specify a file name as the argument for any command that usually requires a sound path. Audio Xtra will look for the sound in the specified sound directory. In authoring, use of a sounds directory is optional. In Shockwave, you must prompt the user for a soundsDirectory before working with any sound on disk.

In authoring or projectors you can set the "soundsDirectory" environment property directly with axSetEnvironmentInfo. In Shockwave you must prompt the user to set the property, and once the directory is set you can not read the path with axGetEnvironmentInfo, although Audio Xtra can use it to read and write files. See the Shockwave security section for an overview of working with Audio Xtra in that environment.

Example:

```
userResponse = axPromptForSoundsDirectory()
```

axCompressToOggFile(soundNameString, filePathString, soundQuality) where soundNameString is the name of a sound present in the sound list, filePathString is the file name to save the new file and soundQuality is a number from 1 to 10 that specifies the compression quality target used by the Ogg Vorbis encoder. Returns an error number or 0 for no error. The last parameters is optional, and if it is not included the encoder will use quality level 4, which equates roughly to 128Mb/s VBR compression. This funcion saves an existing sound in the sound list to a compressed file on the hard drive. The compressed sounds has the same sample rate and number of channels as the original sound, and 16 bit as the sample size. Does not add the sound to the sound list. Use axLoadOggSoundIntoRAM to add the new sound to the sound list. This function was introduced in AudioXtra 6.1.

Limitations

The Vorbis encoder needs to create temporary files in the soundsDirectory, so make sure you set it using either axSetEnvironmentInfo or axPromptForSoundsDirectory before attempting the compression.

Example:

```
-- Compresses existing sound "Recorded" in sound list to

-- the file "PRACTICE.OGG" using default settings

axPromptForSoundsDirectory()

axCompressToOggFile ( "Recorded", "PRACTICE.OGG")

-- Forces conversion to maximum compression

axCompressToOggFile ( "Recorded", "PRACTICE.OGG", 1)

-- Forces conversion to maximum quality

axCompressToOggFile ( "Recorded", "PRACTICE.OGG", 10)
```

axCompressToOggData(soundNameString, soundQuality) where soundNameString is the name of a sound present in the sound list and soundQuality is a number from 1 to 10 that specifies the compression quality target used by the Ogg Vorbis encoder. Returns a property list compatible with ShockFiler's support for sending vList binary data files. The last parameter is optional, and if it is not included the encoder will produce use quality level 4, which equates roughly to 128Mb/s VBR compression. The compressed sounds will have the same sample rate and number of channels as the original sound. This function was introduced in Audio Xtra 6.1, and provides an optimal way for greatly reducing the size and preparing sound files to be uploaded with ShockFiler.

Limitations

The Vorbis encoder needs to create temporary files in the soundsDirectory, so make sure you set it using either axSetEnvironmentInfo or axPromptForSoundsDirectory before attempting the compression.

Example:

axSetEnvironmentInfo("useTempDirectory", 1)

-- Forces conversion to maximum compression

dataToSend = axCompressToOggData ("Recorded", 1)

remoteFileName = "recordedsound.wav"

remoteFileType = #bina

remotefileProperties = []

doappend = 0

showDialog = 1

dialogTitle = "Sending sound file ..."

error = sf_Send ("host", "directory", "usersname", "password" ,dataToSend, '', remoteFileName, remoteFileType, remotefileProperties, doappend, showDialog,dialogTitle)

axCompressToOggByteArray(soundNameString, soundQuality) where soundNameString is the name of a sound present in the sound list and soundQuality is a number from 1 to 10 that specifies the compression quality target used by the Ogg Vorbis encoder. Returns a ByteArray object containing the binary data for the compressed file, which can be used with postNetByteArray to upload the recording to an online server. The last parameter is optional, and if it is not included the encoder will produce use quality level 4, which equates roughly to 128Mb/s VBR compression. The compressed sounds will have the same sample rate and number of channels as the original sound. This function was introduced in Audio Xtra 6.2, and requires Director 11.5, as ByteArray objects are not available in previous versions of Director. Users of Director 11 or earlier can use axCompressToOggBase64.

Limitations

The Vorbis encoder needs to create temporary files in the soundsDirectory, so make sure you set it using either axSetEnvironmentInfo or axPromptForSoundsDirectory before attempting the compression.

Example:

axSetEnvironmentInfo("useTempDirectory", 1)

dataToSend = axCompressToOggByteArray ("Recorded", 1)

```
put dataToSend

-- <ByteArrayObject length = 47655 ByteArray = 0x4f, 0x67, 0x67, 0x53, 0x0,

0x2, 0x0, 0x0, 0x0, 0x0>
```

axCompressToOggBase64(soundNameString, soundQuality) where soundNameString is the name of a sound present in the sound list and soundQuality is a number from 1 to 10 that specifies the compression quality target used by the Ogg Vorbis encoder. Returns a property list containing two properties: #data contains the compressed file encoded as a Bsae64 string, and #length indicates the size of the string. The data portion can be used with postNetText to upload content to an online server, and the property list is formatted in a way that is compatible with both vList and ShockFiler as well. The last parameter is optional, and if it is not included the encoder will produce use quality level 4, which equates roughly to 128Mb/s VBR compression. The compressed sounds will have the same sample rate and number of channels as the original sound. This function was introduced in Audio Xtra 6.2.

Limitations

The Vorbis encoder needs to create temporary files in the soundsDirectory, so make sure you set it using either axSetEnvironmentInfo or axPromptForSoundsDirectory before attempting the compression.

Example:

```
axSetEnvironmentInfo("useTempDirectory", 1)

dataToSend = axCompressToOggBase64 ("Recorded", 1)

pNetID = postNetText("http://myserver.com/page.cgi" , \

[ "Filename": "myfile.ogg", "Base64Data": dataToSend.data])"
```

axLoadOggSoundIntoRAM(soundNameString, fileNameString) where soundNameString is the name for the sound in the sound list and fileNameString is the file name for the external compressed Ogg Vorbis sound file to be loaded. Returns an error number or 0 for no error.

This function converts an external Ogg Vorbis file into WAVE format, and puts it into Audio Xtra's sound list so that it can work with the file, which is automatically loaded into memory. A sound loaded in to RAM will play back smoothly.

This function was introduced in AudioXtra 6.1.

Limitations

The Vorbis decoder needs to create temporary files in the soundsDirectory, so make sure you set it using either axSetEnvironmentInfo or axPromptForSoundsDirectory before attempting to load a file.

Example:

axPromptForSoundsDirectory()

-- loads BIRD.OGG from the sounds directory

axLoadOggSoundIntoRAM("Bird Song","BIRD.OGG")

axLoadOggByteArrayIntoRAM(soundNameString, ByteArrayObject) where soundNameString is the name for the sound in the sound list and byteArrayObject is a Director 11.5 ByteArray object containing the binary Ogg data to be loaded. Returns an error number or 0 for no error.

This function converts Ogg Vorbis data stored in a ByteArray into WAVE format, and puts it into Audio Xtra's sound list so that it can work with the file, which is automatically loaded into memory. A sound loaded in to RAM will play back smoothly.

This function was introduced in AudioXtra 6.2, and requires Director 11.5, as ByteArray objects are not available in previous versions of Director. It can be used with axCompressOggToByteArray to transfer compressed sounds to and from an online server using postNetByteArray and getNetByteArray operations.

Limitations

The Vorbis decoder needs to create temporary files in the soundsDirectory, so make sure you set it using either axSetEnvironmentInfo or axPromptForSoundsDirectory before attempting to load a file.

Example:

axPromptForSoundsDirectory()

axLoadOggByteArrayIntoRAM("Bird Song", myByteArray) axLoadOggBase64DataIntoRAM(soundNameString, base64string) where soundNameString is the name for the sound in the sound list and base64string is a Lingo string containing an Ogg Vorbis sound file with Base64 encoding. Returns an error number or 0 for no error.

This function converts Ogg Vorbis data encoded as a Base64 string into WAVE format, and puts it into Audio Xtra's sound list so that it can work with the file, which is automatically loaded into memory. A sound loaded in to RAM will play back smoothly.

This function was introduced in AudioXtra 6.2, and can be used with axCompressOggToBase64 to transfer compressed sounds to and from an online server using postNetText and getNetText operations.

Limitations

The Vorbis decoder needs to create temporary files in the soundsDirectory, so make sure you set it using either axSetEnvironmentInfo or axPromptForSoundsDirectory before attempting to load a file.

Example:

```
axPromptForSoundsDirectory()

axLoadOggBase64DataIntoRAM("Bird Song", myBase64string)
```

AUDIO XTRA HELP: ERROR CODES

The following is a list of error codes returned by Audio Xtra. It is important to notice that negative error codes are not Audio Xtra errors, but Director standard error codes.

0	No error
-8219	Unable to get file path. File not found.
-8220	Invalid file operators
-8221	File not found
-8222	File positioning problem
-8223	Disk is full
-8224	File permission. Device in use.
-8225	Out of file handles. Current record settings may not be supported by the installed audio driver.
-8226	Misc. file error
-8227	File already exists
-8228	Lost network connection
-8229	File is not open
-8230	Chunk not found
-8231	Invalid swap record
-8232	At end of file
-8233	Bad chunk size
-8234	Drive not found
-8235	File path is too long
-8236	Invalid directory
-8700	Out of memory
-8701	Bad sound name

-8702	Bad data type
-8703	Misc. sound error
-8704	Bad sound type
-8705	Invalid member type
-8706	Bad info type
-8707	Duplicate sound name
-8708	MIX services unavailable
-8709	Cannot read this file format
-8710	File format not supported
-8711	Cannot set this
-8712	No platform support
-8713	No Shockwave permission
-8714	Bad number of arguments
-8715	Value out of range
-8716	Too much hard drive space
-8717	Cannot show Shockwave permissions
-8718	This feature is not supported in Shockwave
-8719	No private directory has been specified
-8720	Invalid file path (contains path delimiters)
-8721	Device not found
-8724	unable to open mixer
-8725	Unable to get mixer information
-8726	No wave input devices
-8727	Cannot control volume for this device
-8728	Cannot control meter level for this device
-8729	Misc. mixer error
-8730	This is a mono device (used when trying to set/get the right volume)

-8731	Mixer is unable to select a device
-8800	Sound recorder closed
-8801	Sound recording device in use
-8802	Sound recorder busy
-8803	Invalid sample rate
-8804	Invalid sample depth
-8805	Invalid compression type
-8806	Input source string not found
-8807	Input source type not found
-8808	Sound hardware error
-8809	No sound is recording
-8810	Zero length sound recording
-8811	Invalid file format
-8812	Recorder already opened
-8813	Zero-length recording buffer
-8814	No sound input device available for recording. Verify that the sound card is functioning and its driver is installed.
-8815	Invalid number of channels.
-8816	Invalid sound format settings.
-8817	RAM buffer exceeded.
-8818	Recording is not paused (tried to call resume)
-8819	Recording is already paused (tried to call pause)
-8820	RAM recording buffer is full
-8821	Invalid file extension for platform
-8900	Sound already playing
-8901	Sound not playing
-8902	No sounds in list

-8903	Zero length sound
-8904	No free sound channels
-8905	Bad sound format data
-8906	Can not delete: sound is busy
-8950	Width out of range (min = 4, max = 3096)
-8951	Height out of range (min = 4, max = 255)
-8952	Sound length is zero
-8953	Sample frames are zero
-8954	Channel is out of range (must be at least 1, max is # of channels in the sound)
-8955	Plot buffer size is zero
-8956	Plot buffer size not correctly divisible
-8957	Plot buffer size too small
-8958	Plot resolution is not one of the valid symbols (#low,#medium,#high,#maximum)
-8959	Resolution too high for the width specified Will happen, if, for example, you try to plot a 100 ms sound over a width of 3000 pixels -- not enough sound data for this. Try #maximum if you get this or else decrease the width of the bitmap
-9000 to -9404	Internal Audio Xtra error code. Should never be returned. Contact support with the name of the function that produced the error.
-9406 on	

AUDIO XTRA HELP: USING THE XTRA IN SHOCKWAVE

Audio Xtra can be used in Shockwave: the distribution package provides packaged Xtras that are downloaded automatically to the user's machine, and installed on demand. Please consult Macromedia's web site for a complete overview of the Xtras automated download mechanism: read the Shockwave Xtras downloading overview technote. The basic steps required to make Audio Xtra available for download are outlined below.

To create a Shockwave movie that will auto-download the Xtra to the user's hard drive you must do the following, in this order:

1. Upload the packaged Xtra files to your web server

2. Modify the entry for Audio Xtra in file xtrainfo.txt to point to the packaged Xtra files on your server

3. Do Modify -> Movie -> Xtras, select Audio Xtra, and check the "Download if Needed" option

Once you have completed steps 1 and 2, you can create other Shockwave movies by doing only step 3.

PACKAGED FILES

Your Audio Xtra archive contains a subfolder called "Shockwave". There are four files inside it:

AudioXtra.w32 - Win 32 packaged Audio Xtra

AudioXtra.ppc - Mac Classic package

AudioXtra.carb - Mac OSX Carbon package

AudioXtra.xpku - Mac OSX Universal Binary package

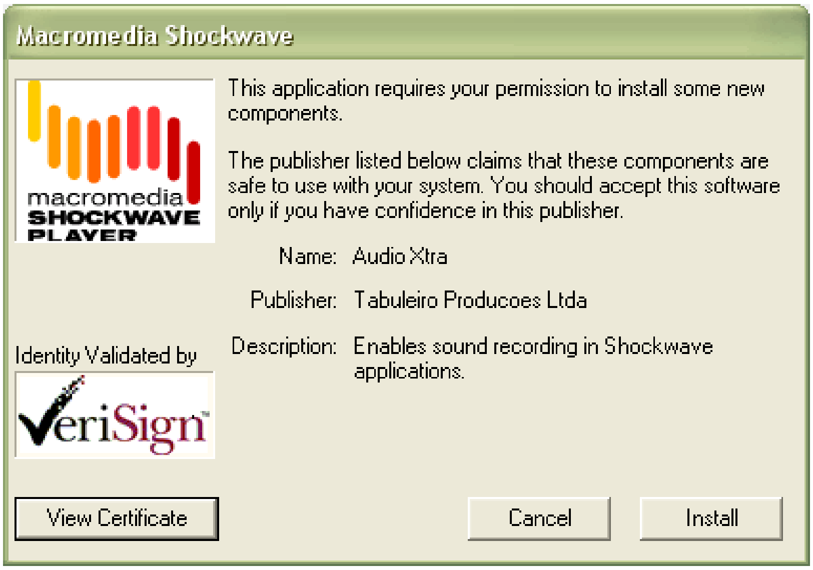
All packages contain the Audio Xtra for that platform. Depending on the user's platform, a package autodownloaded to the user's hard drive will install the correct Audio Xtra for the user's platform into their Shockwave support folder.

If, for some reason, you choose not to make your Shockwave movies autodownload the package files, you can have the user install the right Xtra for their platform into the Shockwave support folder manually.

Upload all package files to the same directory on your web server. Use a "binary" or "raw", not "text" transfer. If the packages are uploaded to two different directories, autodownloading will not work. Do not rename the package files.

If you are going to distribute Audio Xtra with Shockwave movies, we recommend that you use your own web server to do so. Packages are available at Tabuleiro's download services, but we reserve the right to refuse access, without notice, to any referring URL that generates excessive traffic.

The Audio Xtra packages included with the download have been signed and packaged by Tabuleiro, and will present the following security message to users of your Shockwave movies when they are installed for the first time:



Macromedia Shockwave

This application requires your permission to install some new components.

The publisher listed below claims that these components are safe to use with your system. You should accept this software only if you have confidence in this publisher.

Name: Audio Xtra

Publisher: Tabuleiro Producoes Ltda

Description: Enables sound recording in Shockwave applications.

You may choose to repackage Audio Xtra and sign it with your own Verisign certificate. You might want to do this if you want your own company name to appear in the auto-download dialog box the user sees when an auto-download is initiated. Adobe is the best source of information on applying for a Verisign certificate and packaging files.

XTRAINFO.TXT

The text file xtrainfo.txt resides in your Director authoring directory. It contains information about Xtras such as file version names for an Xtra on both platforms and the URL for the packages. The information contained in xtrainfo.txt is saved with each movie you create and used by projectors and Shockwave.

You must create an entry for Audio Xtra in your xtrainfo.txt file that specifies the URL on your server for the package files. The last part of the path will always be "AudioXtra". That specifies the filename of the packages within the directory, without the file extension. Do not include a file extension at the end of the path.

```
[#namePPC:"Audio Xtra", #nameW32:"AudioXtra.x32",
#package:"http://www.domain.com/folder/AudioXtra"]
```

Make sure that the line above does not contain any return character after you paste it into your xtrainfo.txt file. Open your text editor wide and make sure the line does not wrap. If the opening and closing brackets are not on the same line, Director will not be able to create a valid list from the entry and the "Download if needed" button will be dimmed for Audio Xtra in Director.

If you edit xtrainfo.txt while Director is open you should quit and restart Director to read in the changed information in xtrainfo.

EDITING THE MOVIE'S XTRAS LIST

Open the Director movie that you want to save as Shockwave. Choose Modify -> Movie -> Xtras and add Audio Xtra. Select Audio Xtra from the list and check the "Download if needed" option. Director will initiate an internet connection and look for the packages at the URL you specified in xtrainfo.txt.

If Director finds the packages, it will transfer information about the package contents for both platforms such as file names and version numbers and embed the information into your Director movie. An informational dialog box will appear that tells you that the packages for both platforms are "downloading". The packages themselves are not downloading, just information about them that the Shockwave movie will need later to compare the version of the Xtra the user possibly already has to the version currently on the server in order to determine if autodownloading is necessary. The Director movie needs information about both platforms because it may find itself running on either platform once it is on the web.

Once "downloading" of the packages has finished, save the movie, then publish as Shockwave. The finished Shockwave movie can reside at any URL. It does not have to be in the same directory or even on the same server as the packaged Xtras.

If a connection cannot be opened, or the packages cannot be found at the specified location, Director will uncheck the "Download as needed" option automatically. You must have a successful connection for the box to remain checked. A Shockwave made out of a Director movie with the "Download as needed" button unchecked will not autodownload Audio Xtra.

MACROMEDIA XTRAS THAT AUDIO XTRA NEEDS

Audio Xtra uses some services provided by Macromedia Xtras. Depending on what Audio Xtra operations you are going to perform you must include at least four of the following Xtras in your movie's Xtras list and possibly all 5:

Mix Services: always needed

MacroMix: always needed

DirectSound: always needed

Sound Import Export: always needed

Sun AU Import Export: optionally needed to record and play Sun AU files

All of the Xtras can download if needed and their paths are already set up in Director's xtrainfo.txt. Some of them are automatically installed with the standard Shockwave install. To include them in your movie choose Modify -> Movie -> Xtras and check "Download if needed". A web connection must be open for the Xtras to be added successfully.

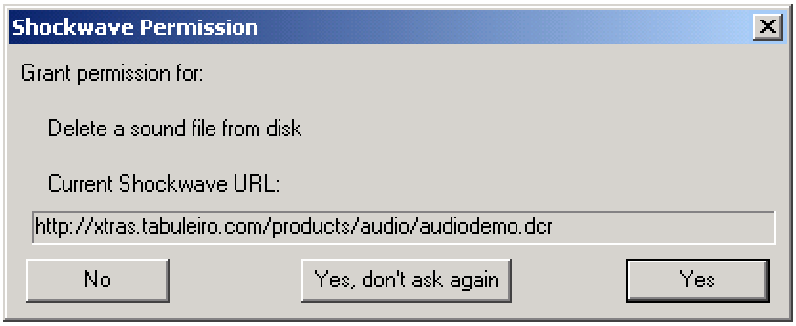
SECURITY

Audio Xtra records files to the user's hard drive and queries the system for information that it needs, such as writeable hard drives. In a Shockwave environment the user must be informed of such activity and given an opportunity to grant or refuse permission. Audio Xtra restricts access to certain functions and system properties when running in Shockwave. For testing, you can simulate a Shockwave environment in authoring by doing:

```
axSetEnvironmentInfo("simulateShockwave",TRUE)
```

OBTAINING USER PERMISSION

Some Audio Xtra commands display a permission dialog box in Shockwave when they are issued. The dialog box looks like this:



Shockwave Permission

Grant permission for:

Delete a sound file from disk

Current Shockwave URL:

http://xtras.tabuleiro.com/products/audio/audiodemo.dcr

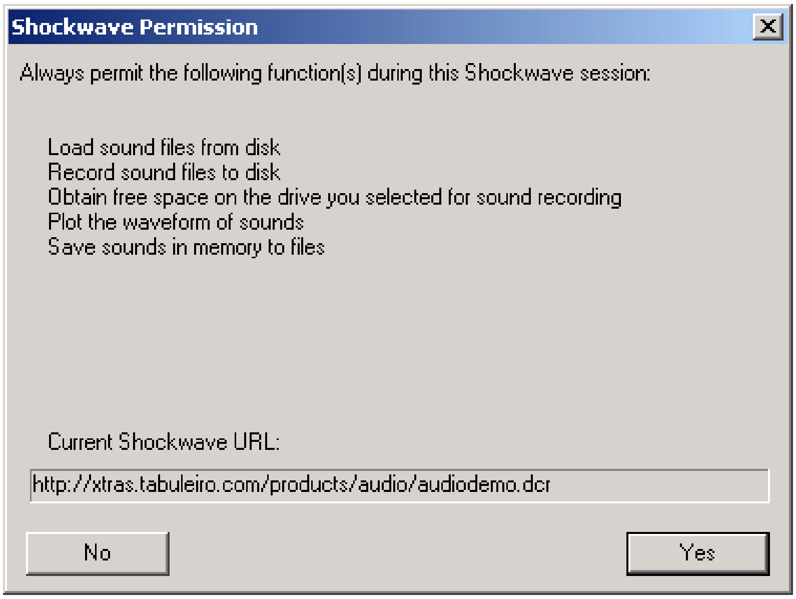
This is what happens, depending on the user's response:

No: command returns error -8713, "No Shockwave permissions"

Yes, don't ask again: command is carried out and user will not be prompted for permission for that command again during the session unless the Shockwave movie changes

Yes: command is carried out, but user will be prompted again the next time the command is issued

Audio Xtra's default permission behavior is to display the permission dialog every time a command needing permission is issued. You can head off the default behavior by using axAskPermission to display a permission dialog early on, listing the commands you plan to use during the session. Once the user grants permission in this dialog the user is no longer prompted again during the session for the commands that were displayed in the dialog, unless the Shockwave movie changes. This is what the dialog displayed by axAskPermission looks like, although the list of commands will differ depending on what commands you have requested it to list:



Shockwave Permission

Always permit the following function(s) during this Shockwave session:

Load sound files from disk
Record sound files to disk
Obtain free space on the drive you selected for sound recording
Plot the waveform of sounds
Save sounds in memory to files

Current Shockwave URL:

http://xtras.tabuleiro.com/products/audio/audiodemo.dcr

This is what happens, depending on the user's response:

No: command returns error -8713, "No Shockwave permissions"

Yes: command is carried out and user will not be prompted for permission for the listed command(s) again during the session unless the Shockwave movie changes

PRIVATE SOUNDS DIRECTORY

In Shockwave you must have a sounds directory specified before you can record audio files. In authoring or projectors you can specify any path for a recorded sound and load files from any path. If you decide to use a sounds directory in authoring or projector for convenience you can either set the path to the sounds directory using axSetEnvironmentInfo ("soundsDirectory") or you can prompt the user for the path using axPromptForSoundsDirectory.

In Shockwave, there are two options: the first is to prompt the user for the path using axPromptForSoundsDirectory. The second option is to set the environment option "useDswMediaDirectory" to TRUE, and Audio Xtra will automatically detect and use the DswMedia support folder for your Shockwave player installation. This option was introduced in version 5.1 of Audio Xtra.

In both cases you cannot set or query the "soundsDirectory" property using axSet or GetEnvironmentInfo, for security reasons. The path to the final sounds directory is stored internally and only available to Audio Xtra. The Shockwave movie can only work with files in the sounds directory. This applies to recording, playback and sound conversion.

SYSTEM INFORMATION

Some system properties returned from axGetEnvironmentInfo are not available when running under Shockwave. They return error -8718, "This feature is not supported in Shockwave". The properties not available in Shockwave are:

tempPath

driveList

writeDriveList

soundsDirectory

AUDIO XTRA HELP: FILE TYPE EXTENSIONS

On the Windows platform you must include a file type extension at the end of the file name when recording to a file or converting a sound to a file. The file type extension must match the sound type. On the Mac, adding a file type extension to the file name is optional.

File type	Extension
Microsoft WAVE	.wav
Sun AU	.au
Apple AIFF	.aif

On both platforms, Audio Xtra tries to interpret a sound file in the format specified by the extension. If a sound file name has the wrong extension, for instance if an AIFF file is named SOUND.WAV, Audio Xtra will not be able to play or otherwise work with the file.

AUDIO XTRA HELP: CREATING PROJECTORS

Audio Xtra can be used to create projector in all operational systems and platforms supported by Director 8.5, Director MX, Director MX 2004 and Director 11.

CREATING A CROSS-PLATFORM OR STANDARD PROJECTOR - Director MX 2004 and Director 11

These versions of Director include the ability to create Windows projectors when hosted on Mac OS X, and vice-versa. Director MX 2004 can create projectors for older versions of MacOS (8 and 9) as well. However, this only works correctly if Director is configured to locate and include the proper Xtra files for "the other" platform. Please make sure the appropriate files are installed in the Configuration\Cross platform resources Director folder according to the installation notes, and also make sure the Configuration\xtrainfo.txt file contains information about Audio Xtra (this procedure is also covered in the installation instructions)

Audio Xtra uses Macromedia Xtras to provide some additional import/export services, so make sure all the Xtras necessary for proper operation of the Audio Xtra are also included in your movie. You must include the following Macromedia Xtras in your Director movie Xtras list or in the projector Xtras folder, depending on what sound file formats your projector will be working with:

Audio Xtra : always needed

Mix Services: always needed

MacroMix: always needed

DirectSound: always needed

Sound Import Export: always needed

Sun AU Import Export: optionally needed to record and play Sun AU files

With your .dir file opened, please select the MODIFY->MOVIE->XTRAS menu item. Click the ADD button , and include the Audio Xtra (AudioXtra.x32 on Windows) in the first movie used in your projector. You can then proceed and include the remaining Xtras listed above, if they are not already included. If you plan to use the Xtra in Shockwave you can also check the DOWNLOAD IF NEEDED checkbox for each Xtra. More information about this procedure can be found at the USING THE XTRA IN SHOCKWAVE section of the documentation.

Save the file and the changes will be recorded to your movie.

Assuming the cross-platform files are already installed and configured correctly you can now invoke the FILE->PUBLISH SETTINGS menu item to configure the parameters for your projector. You can then configure what types of projectors will be created, and you can also create a Shockwave version of your file if necessary.

In the FILES tab you can add additional Director movies to your projector. To finalize just click the PUBLISH button. Director saves the publishing settings with your Director movie, and future projectors can be created simply by choosing the FILE->PUBLISH menu item.

CREATING A STANDARD WINDOWS OR MACOS PROJECTOR - Director 8.5 and MX

Director 8.5 and MX can only create native projectors. Windows projectors need to be created on a Windows machine, and Macintosh projectors need to be created on a Macintosh computer. However, Director movies (.dir files) containing Audio Xtra scripts don't need any modification in order to work on both platforms, provided your scripts already take care of selecting the appropriate file extensions and file formats if necessary. A developer can work primarily on the Mac and only transfer the final .dir file to Director for Windows in order to create a Windows projector, or vice-versa. The only requirement is to install the Xtra on both platforms. Instructions for installing the Xtra can be found in the download packages. Please notice that Audio Xtra serial numbers are cross-platform: you can use the same serial number with the axRegister() function on your movie to register the software on both Mac and Windows.

Audio Xtra uses Macromedia Xtras to provide some additional import/export services, so make sure all the Xtras necessary for proper operation of the Audio Xtra are also included in your movie. You must include the following Macromedia Xtras in your Director movie Xtras list or in the projector Xtras folder, depending on what sound file formats your projector will be working with:

Audio Xtra : always needed

Mix Services: always needed

MacroMix: always needed

DirectSound: always needed

Sound Import Export: always needed

Sun AU Import Export: optionally needed to record and play Sun AU files

With your .dir file opened, please select the MODIFY->MOVIE->XTRAS menu item. Click the ADD button , and include the Audio Xtra (AudioXtra.x32 on Windows) in the first movie used in your projector. You can then proceed and include the remaining Xtras listed above, if they are not already included. If you plan to use the Xtra in Shockwave you can also check the DOWNLOAD IF NEEDED checkbox for each Xtra. More information about this procedure can be found at the USING THE XTRA IN SHOCKWAVE section of the documentation. Save the file and the changes will be recorded to your movie. Now you just need to select CREATE PROJECTOR from the FILE menu in Director to create your projector, and all the necessary files will be included in it.

TIP: You can also deliver the required Xtras in a folder named XTRAS, located in the same directory of your Projector, if you do not want to embed the Xtra file into your projector. This is recommended for faster startup of your program.

AUDIO XTRA HELP: SOUND LIST TABLE

The Table below lists the properties stored in the Sound List. Note that some properties cannot be set.

Information Stored in Sound List

	Data			
Property	Type	Meaning	Get	Set
name	string	name of the sound	X	X
source	string	where the sound is from ("file", "RAM" or "member")	X	
format	string	sound file format	X	
length	integer	length of sound in milliseconds	X	
position	integer	current playback position of sound in milliseconds. Property will be 0 unless sound is currently playing.	X	
depth	integer	sample bit depth of the sound, for instance 8 or 16	X	
rate	integer	sample rate of the sound, for example 22050	X	
volume	integer	playback volume of the sound in range 0 - 255. The volume of the Director sound channel this sound plays back in will be set to this value.	X	X
channel	integer	playback channel of the sound (0 if not playing)	X	
numChannels	integer	number of channels	X	

AUDIO XTRA HELP: ENVIRONMENT PROPERTIES TABLE

The Environment Properties Table below lists the available properties. Note that some properties only apply to one platform and some cannot be set.

Environment Properties

Info	Data type	Meaning	Platform				Get/Set	
overwrite	boolean 0 or 1	If TRUE (1) will overwrite an existing member or sound file if it is specified as the target to record to. If FALSE it will not overwrite an existing member or file and the command attempting to will return an error. Default is TRUE.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
inputVolumeGain	integer from 63-255	Mac recording input volume. Default is 159. On the Mac it can never be turned off completely. On some systems there will be less than 255 discrete settings. On those systems setting a value will set it to the closest valid value. To set recording volume on the PC you must set it for the Windows Mixer Device using the inputSourceVolume property.		Mac Classic	Mac OSX	Shock wave	Get	Set
recordSampleRate	integer sample rate	Sample rate for recording. For example 44100 specifies 44.1 khz. Standard rates are: 11025, 22050, 44100, 48100.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
recordSampleDepth	integer	Sample bit depth for recording. Typically 8 or 16. Playback of a sound recorded at a non-standard depth will not work unless the depth if supported by the sound card.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
recordNumChannels	integer	Number of channels to record. Device must support stereo	Win	Mac Classic	Mac OSX	Shock wave	Get	Set

		recording for this to work when set at 2. Setting this to 1 on Mac for a device that doesn't support it will return a hardware error.						
outputDefaultVolume	integer from 0-255	Default volume for sound playback in Director. Reflected in the sound property "volume" when sound is viewed in sound list. Can be overridden by setting the property for an individual sound via axSetSoundInfo.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
soundManagerVersion	string	Version of the Mac Sound Manager in use.		Mac Classic	Mac OSX	Shock wave	Get	
totalChannels	integer	Total number of channels on the system available for sound playback	Win	Mac Classic	Mac OSX	Shock wave	Get	
freeChannels	integer	Number of available channels (not already in use) for sound playback	Win	Mac Classic	Mac OSX	Shock wave	Get	
playthrough	boolean 0 or 1	Whether or not microphone playthrough is on. Playthrough plays the mike or other input device input through the speakers as sound is being recorded. Changing the playthrough on some systems may change the current input device. Note: On Mac, set and get this property only while the recorder is open, otherwise the system may experience errors.		Mac Classic	Mac OSX	Shock wave	Get	Set
inputDevice	string	The name of the current input device used for recording. On most machines there is only one input device, usually a single sound card. But some systems have multiple devices, including multiple sound cards, USB microphones and WebCams. The Xtra will automatically identify and use the preferred audio device for voice recording at startup, according to system	Win	Mac Classic	Mac OSX	Shock wave	Get	Set

		preferences.						
inputDeviceList	string	Return-delimited list of all available input devices on the system. Most machines have only one input device, a single sound card.	Win	Mac Classic	Mac OSX	Shock wave	Get	
preferredRecordingDevice	string	The name of the preferred audio device for audio recording on the system. On Windows this information is obtained from the registry, reflecting the changes made by the user with the Sounds Control Panel. On the Mac the preferred device is also set in the Sounds Control Panel, or System Preferences -> Sound under OSX.	Win	Mac Classic	Mac OSX	Shock wave	Get	
preferredVoiceRecordingDevice	string	The name of the preferred audio device specified for voice recording on the system. This can be configured separately from the preferred audio recording device on Windows XP, in the Sound and Audio / Voices control panel. On previous Windows versions and on the Mac this property returns the same value as the preferredRecordingDevice string.	Win	Mac Classic	Mac OSX	Shock wave	Get	
inputSource	string	Input source in the current input device that will be used for recording. Each input device can have one or more sources. On Windows the sources are usually input connections in the sound card (microphone, line in, etc.) On the Mac the number of input sources depends on the device used: internal sound cards may have multiple sources (internal mic, external mic, etc.). Other devices like USB microphones have only one source.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
inputSourceList	string		Win				Get	

		Return-delimited list of all available input sources on the current input device.		Mac Classic	Mac OSX	Shock wave		
availableSoundFormats	string	A return-delimited list of Director-supported sound formats for playback. The formats are supplied by the available MIX Xtras. Audio Xtra currently supports a subset of the available formats.	Win	Mac Classic	Mac OSX	Shock wave	Get	
availableConversionFormats	string	A return-delimited list of Director-supported sound formats for exporting files. The formats are supplied by the available MIX Xtras. Audio Xtra currently supports a subset of the available formats.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
recordingBufferSize	integer	The size of the recording buffer. Set by axOpenRecorder(). Recorder must be open or it will return an error.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
mixAvailable	boolean 0 or 1	TRUE if the MIX Services Xtra is loaded. If this property is FALSE, many Audio Xtra functions will fail. Debug aid.	Win	Mac Classic	Mac OSX	Shock wave	Get	
tempPath	string	System temp directory. You can use this property to set the sounds directory to the user's temp folder. The path is returned in DOS format, which converts long file names to an 8.3 format. Both Director and Audio Xtra recognize DOS format file paths.	Win	Mac Classic	Mac OSX		Get	
driveList	string	List of all local and network drive/volume names on the system including removeable and CD. On Win the network drives returned are only those that are mapped to a drive letter.	Win	Mac Classic	Mac OSX		Get	
writeDriveList	string	List of only writeable drive names. NOTE: Although	Win	Mac Classic	Mac OSX		Get	

		writeable network drives are listed, recording to a network drive may cause skips in the audio if the network throughput cannot keep up with the volume of recorded data.						
useDswMediaDirectory	boolean 0 or 1	This property works under Shockwave ONLY, and accepts a boolean parameter that indicates if the Xtra should attempt to create a DswMedia folder if one does not exist. When the parameter is set to FALSE the Xtra will attempt to use the default DswMedia folder if it is available at the default location. If this fails it will also look for a DswMedia folder in the same directory as the XTRAS folder, to match Shockwave 8 installations. However, in some cases no DswMedia folder is present, and the function will return an appropriate error message (file not found.) If necessary you can check the error message and call the function again with the creation parameter set to TRUE. In this case the Xtra will attempt to locate the DswMedia folder AND create a new one at the default location if it does not exist. Check the Audio Xtra sample movie for an example on how to use this property effectively and test for the error codes. This property is new in Audio Xtra 5.1.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
useTempDirectory	boolean 0 or 1	When this property is set, Audio Xtra will set the sounds directory to use a system temporary path for all	Win	Mac Classic	Mac OSX	Shock wave	Get	Set

		operations that involve file recording and conversion. This property is new in Audio Xtra 6.						
simulateShockwave	boolean 0 or 1	If TRUE will simulate Shockwave behavior in authoring. Always returns TRUE when running in Shockwave and cannot be reset.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
soundsDirectoryIsSet	boolean 0 or 1	TRUE if the default sounds directory has been set with axPromptForSoundsDirectory or via the environment property useDswMediaDirectory	Win	Mac Classic	Mac OSX	Shock wave	Get	
soundsDirectory	string	Path to sounds directory	Win	Mac Classic	Mac OSX		Get	Set
peakVolume	integer	Available only when a recording is taking place. Returns the peak volume for the last recording buffer. Range 0-127.	Win	Mac Classic	Mac OSX	Shock wave	Get	
inputSourceVolume	integer	Sets the volume of the selected Windows input source. Range 0-255.	Win			Shock wave	Get	Set
inputSourceLeftVolume	integer	Sets the left channel volume of the selected Windows input device. Only works for sound devices that support stereo volume.Range 0-255.	Win			Shock wave	Get	Set
inputSourceRightVolume	integer	Sets the right channel volume of the selected Windows input device. Only works for sound devices that support stereo volume.Range 0-255.	Win			Shock wave	Get	Set
plotBufferSize	integer	The size in bytes of the RAM buffer to be used with axPlotWaveform. The larger the buffer, the faster the wave-plotting. If the buffer is smaller than the sound data to process, the buffer will be re-used to process chunks of sound until plotting is finished.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
convertSampleRate	integer		Win				Get	Set

		Specifies the sample rate of the destination file or castmember when calling axConvertToFile or axConvertToMember. Set this property to 0 to take the default, which is usually the same as the sample rate of the source sound if the format you are saving to supports that rate. Maximum value this property can be set to is 88200.		Mac Classic	Mac OSX	Shock wave		
convertSampleDepth	integer 8 or 16	Specifies the sample bit depth of the destination file or castmember when calling axConvertToFile or axConvertToMember. Set this property to 0 to take the default, which is usually the same as the sample depth of the source sound if the format you are saving to supports that rate. 8 and 16 are the only legal values for this property.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set
convertNumChannels	integer 1 or 2	Specifies the number of channels in the destination file or castmember when calling axConvertToFile or axConvertToMember. Set this property to 0 to take the default, which is usually the same as the number of channels of the source sound if the format you are saving to supports that value. 1 and 2 are the only legal values for this property.	Win	Mac Classic	Mac OSX	Shock wave	Get	Set

AUDIO XTRA HELP: HOW TO ORDER & REGISTER

The unregistered version of Audio Xtra is fully-functional and may be used for evaluation, nonprofit and educational purposes only: commercial distribution is strictly prohibited. A registered version of the Xtra can be used in commercial products, and may be purchased online at xtras.tabuleiro.com, using a secure server. At our web site you can also consult our purchase policy, purchase instructions, payment, delivery and security methods.

If you decide to buy the Xtra you don't need to download a new copy of the software. After your order is processed you will receive an e-mail with a serial number to register the software you've already installed on your machine.

To register the Xtra you should use the axRegister() function, usually called at the startup of your movie, or before an Audio Xtra function is used. More information about specific syntax can be found at the Xtra Functions page. Please keep your serial number archived for future reference.

AUDIO XTRA HELP: LICENSING & AVAILABILITY

Audio Xtra is a commercial product. Current price and updated information can be found at xtras.tabuleiro.com. If your product provides printed documentation and package we ask you to kindly include the following copyright information:

Audio Xtra(tm) (c) Tabuleiro Prod. Ltda 2008

No royalty-fees are required for a distribution of the Xtra with your product.

Audio Xtra includes Ogg Vorbis encoding libraries, released under the following BSD-style license:

the use of this software, even if advised of the possibility of such damage.

AUDIO XTRA HELP: TECHNICAL SUPPORT

Please use the Your Account section available at our web site xtras.tabuleiro.com to submit your questions. The site also contains Technotes and other resources that can help you identify and solve the most common problems quickly.