

# 1 Introduction

This reference is intended to assist in diagnosing and correcting unusual problems with XStudio Voice Tracker. It covers areas that are not contained in the standard help and manual documentation.



Some techniques involve manually modifying the registry. It is strongly urged that the sections of the registry that pertain to XStudio Voice Tracker be backed up before they are modified.

## 2 Audio Engine

The audio engine is installed as a Windows service and is set up to automatically start when the PC's operating system is loaded. This makes the audio engine available before any user login is performed on the host PC, and opens the possibility for controlling the audio engine from another PC.

XStudio Voice Tracker does not presently make use of the remote access functionality.

### ***Current Audio Engine Version Numbers***

<u>File</u>	<u>Version</u>
AudioEngine.exe	Version 1.90 (Build 20070630 - 17:48)
GenCodec.dll	No version info. Date: 6/28/2007 2:11 PM
NCTAudioCompress.dll	Version 1.7.8.0 Date: 2/11/2003 2:32 PM
aeRemote.dll	Version 1.0.0.1 Date: 6/26/2007 10:26 PM

### 2.1 Manual Installation

The audio engine is normally automatically installed when the XStudio Voice Tracker is installed. Should there be a situation where it is not installed, or there's a need to manually-install the audio engine, here are basic steps that should be followed.



The audio engine cannot be used on Windows 95, Windows 98 or Windows Me. Services can only be installed and used on NT-family operating systems.

1. Files needed:

AudioEngine.exe  
GenCodec.dll  
NCTAudioCompress.dll  
aptX100.dll (ONLY IF apt-X is installed and licensed)  
pcgw32.dll (ONLY IF apt-X is installed and licensed)

2. Copy all needed files to a common folder - all files must be in the same folder. In XStudio Voice Tracker, the normal location for these files is a folder located under the application folder named "**Engine**". An exception is the aptX100.dll file, which can be located in the Windows path, although

not recommended.

3. The NCT dll must be registered with the operating system. To register the dll, open a command window (DOS window) and navigate to the folder in which the NCT dll is located. At the command prompt, enter the following:

```
regsvr32 NCTAudioCompress.dll
```

Press the **[Enter]** key to register the dll.

To unregister the NCT dll, open a command window and type: **regsvr32 /u NCTAudioCompress.dll**, followed by the **[Enter]** key.

4. The audio engine must be installed as a service. You can do this from a command window (DOS window), similar to the NCT dll registration. Open a command window and navigate to the folder in which the audio engine executable is located. At the command prompt, enter the following:

```
audioengine /install
```

Press the **[Enter]** key to install the service.



To manually uninstall the audio engine from a command prompt, type the following:

```
audioengine /uninstall
```

Press the **[Enter]** key to uninstall the audio engine as a service.

### ***Running the Audio Engine in Console Mode***

For troubleshooting purposes, the audio engine can also be run in console mode. Ensure that the audio engine, if installed as a service, is not running before attempting to run it as a console application.

To start the audio engine as a console application, open a command window (DOS window), navigate to the folder containing the audio engine files, and enter the following at a command prompt:

```
audioengine /console
```

Press the **[Enter]** key to launch the audio engine. The audio engine will launch and the command window will display status information. The command prompt disappears.

To stop the audio engine when running as a console application, press **[Ctrl-C]** in the command window. The audio engine will shut down and the command-prompt will reappear.

## 2.2 Registry Information

The audio engine uses the Windows registry to store its settings, including registration information. The base key for these settings is:

**HKEY\_LOCAL\_MACHINE\SOFTWARE\dcsTools\AudioEngine**

There are several sub-keys containing information about audio channels, available devices, and audio stream configuration.

All of the standard settings are automatically set and maintained by the engine with the exception of the registration information, which must be entered in order for the engine to run as a fully-licensed application. As a default, the audio engine will run 5 days from first launch in trial mode. After that, registration is required.

Some of the registry keys can be "tweaked" to correct or further diagnose problems. The following is a list of registry settings used by the audio engine, along with a brief explanation as applicable.

### Audio Engine Registry Settings

<u>Key Name</u>	<u>Default Value</u>	<u>Description</u>
<b>HKEY_LOCAL_MACHINE\SOFTWARE\dcsTools</b>		
CodeTraceLevel	Key does not exist.	DWord. This key is used to force the audio engine to create dump files for tracing codec problems. It is used primarily for determining why apt-X or MPEG errors are occurring, with messages like "codec not available".  When the key is created and assigned a value of <b>2</b> , dump files with the naming syntax <b>GenCodecXXX.dmp</b> are created, with a new <b>XXX</b> value each time the audio engine is launched.
<b>HKEY_LOCAL_MACHINE\SOFTWARE\dcsTools\AudioEngine</b>		
Registration Key	5-Day Trial auto-generated by the audio engine.	REG_SZ. Contains the user-entered registration key, which enables full functionality of the audio engine. The audio engine as a default supports PCM, apt-X and MPEG encoding, although the apt-X dll must be present and available
TraceLevel	Key does not exist.	DWord. This key must be added manually. It is used for troubleshooting and when present, increases the amount of information stored in the <b>AudioEngine.dmp</b> log file. To increase the logging detail, create this key as a <b>REG_DWORD</b> type and enter a value of <b>2</b> .
<b>HKEY_LOCAL_MACHINE\SOFTWARE\dcsTools\AudioEngine\Channels</b>		

<u>Key Name</u>	<u>Default Value</u>	<u>Description</u>
PipeClock	44100	DWord. The clock rate for audio handling. For Windows driver devices, this value can be changed, but for devices like the Digigram audio card, it must be left alone.  Valid values are: 32000, 44100, and 48000
<b>HKEY_LOCAL_MACHINE\SOFTWARE\dcsTools\AudioEngine\Device_x</b>		
(There could be more than one of these entries if a Digigram or AudioScience card is installed in the PC in addition to standard wave devices.)		
DeallocateIdlePipes	Engine-maintained.	DWord = 1
DefaultRecordInput	Engine-maintained.	DWord = 0
EnableWatchdog	Engine-maintained.	DWord = 0
PlayPipeMask	Engine-maintained.	DWord = 4294967295
PreallocatePlayPipes	Engine-maintained.	DWord = 0
PreallocateRecordPipes	Engine-maintained.	DWord = 0
RecordPipeMask	Engine-maintained.	DWord = 4294967295
StreamCount	Engine-maintained.	DWord = 3
WatchdogTimeout	Engine-maintained.	DWord = 500
<b>HKEY_LOCAL_MACHINE\SOFTWARE\dcsTools\AudioEngine\Devices</b>		
Device_1	0	DWord. Device_1 contains wave device information.
Device_2	4294967295	DWord. 2nd device record.
Device_3	4294967295	DWord. 3rd device record.
Device_4	4294967295	DWord. 4th device record.
<b>HKEY_LOCAL_MACHINE\SOFTWARE\dcsTools\AudioEngine\Streams</b>		
BufferCount	8	DWord. The number of buffers allocated for audio data handling. The larger the number of buffers, the more memory consumed. If the BufferSize value is reduced, it may be beneficial to increase this value to something like 12.
BufferSize	32768	DWord. The size of each individual audio data buffer. Valid values can range from a

<u>Key Name</u>	<u>Default Value</u>	<u>Description</u>
		<p>low of 2048 to a high of 32768 (the default).</p> <p>Recommended values in case of a need to change would be on 4K boundaries - ie, 8192, 12288, 16384, 20480, 24576, 28672. General recommendation is to stick with 8K boundaries - 8192, 16384, 32768.</p> <p>This value, by observation, can have an impact on audio play response time. If the play source is a network drive and the buffer is the default, some delay in start of audio may be noticed as the buffer is being filled and decompressed. To improve response, reduce the buffer size. See the topic on <a href="#">slow play response</a>.</p>
MpegCodec	0	
RateConvertQuality	1	
UseMpegCodec	0	

## 2.3 Specific Problems

This section documents specific problems that may occur and the solution and/or work-around. Before applying any of the techniques described, you should check to be sure that you are running the latest version of the application and double-check your application settings (Preferences) to ensure everything is correct.

### 2.3.1 Audio Engine Service Unresponsive

**Symptom:** The audio engine service is started and appears to be running, but client applications are unable to communicate with the audio engine.

Reviewing the application client event log indicates that the audio engine service is running. The aeRemote.dll loads OK according to the event log, but when a connection attempt is made, the attempt times out after looping the attempt 10 times.

**Solution 1:** Check to see if the apt-X encoder is installed with the audio engine and that it is the standard commercial version that requires registration after a 10-day trial period. If so, check to see if the apt-X temporary time period has expired. It can be checked one of two ways:

1. Use the **apt-X Registration Utility** included in audio engine installations - the file name is aptX\_RegDlg.exe.
2. Stop the audio engine service and launch the audio engine from a command window. See the topic on [running the audio engine in console](#) mode for instructions.

If an expired apt-X trial license is the issue, the apt-X registration dialog will appear using either of the two methods. Correct the problem by registering the software or removing it.

### 2.3.2 Slow Playback Response Time

It is possible that a combination of factors comes together to cause what might be called a "hesitation" in playback of audio when rehearsing sequences. If there is a noticeable delay between the time you click on the [Play] button and when you hear audio - on the order of 1/2-second or more, there are some things that can be done to improve response time.

The problem most generally manifests itself when the playback audio folder is on a remote machine - as would be the case with centrally-stored audio.

Here are some steps to try to improve response time - try them in order, shutting down the audio engine before making the changes, then restarting the audio engine and testing the change. It is recommended you make one change at a time to get a good sense of the effect of each change rather than making several changes and not knowing which of them may have fixed the problem.

1. **Change the audio stream buffer size.** The default audio stream buffer size is 32768 (32,768 bytes). Open the registry, locate the key **BufferSize** (see the [Registry Information](#) section for details on its location) and change the value to 8196 (8,196 bytes). Relaunch the audio engine and test the playback start response. If the response is acceptable, shut the audio engine down and increase the value to 16384 (16,384 bytes). Re-test playback response. If it's still acceptable, try increasing the value to 24580 (24,580 bytes) and repeat the test. If playback response time is acceptable, leave the setting. If not, revert the setting back to the last value you tested that produced acceptable results.



Lowering the **BufferSize** value increases hard disk read frequency. The idea is to use the largest value possible while getting acceptable play start response time. This is particularly true of situations where the playback audio is stored

on a central server - you want to keep disk access to a minimum to keep from "hammering" the central storage device any more than necessary.

2. **Change the pipe clock setting.** If you've changed the buffer size down to 8196 and are still not getting acceptable results, you can try changing the pipe clock setting.

**Ensure the audio engine is shut down**, then open the registry and locate the key **PipeClock**. See the [Registry Information](#) section for details on its location. Change the value from it's default of 44100 to 32000, restart the audio engine and test playback response. You can also try setting the value to 48000.

By way of explanation, if the source playback audio was sample at 32kHz and the **PipeClock** setting is 44100 (44.1kHz), the audio engine is rate-converting, which adds some time to the process. The idea is to match this value to the value of the majority of the source audio.



Changing the **PipeClock** value produces best results when you know all or most of the source playback audio was sampled at the same rate. For DCS units, this would be 32000 (32kHz).