



Search

sshdot

3

ie

rail

Research Centers

Core Java

Enterprise Java

Mobile Java

Tools & Methods

JavaWorld Archives

Site Resources

Featured Articles

New s & View s

Community

Java Q&A

JW Blogs

Podcasts

Site Map

Careers

New s letters

Whitepapers

RSS Feeds

About JavaWorld

Advertise

Write for JW

## Java Tip 24: How to play audio in applications

Here's a step-by-step guide to playing audio files in a Java application  
By Chong Ser Wah and John D. Mitchell, JavaWorld.com, 02/01/97

Print

Feedback

22 Comments

What's this?

Playing audio files in Java applications is not officially supported in the current release of Java. But fear not, there is a way! This tip will show you how -- starting with a description of the basic steps involved in playing audio clips in Java applets and then moving on to Java application support.

Playing audio clips in applets is quite simple and involves the following steps:

- Create an AudioClip object
- Load .au sound file into AudioClip
- Play sounds once or loop continuously
- Stop playback

Here's how the code for these steps looks:

```
import java.applet.*;
AudioClip ac = getAudioClip(getCodeBase(), soundFile);
ac.play(); //play once
ac.stop(); //stop playing
ac.loop(); //play continuously
```

It would seem logical to use this same code to play audio clips in a Java application. Unfortunately, if you do that you will get errors from the compiler. Why? Because the AudioClip object and the getAudioClip() method are part of the java.applet package -- and are not part of applications. The good news is we can dive down and make things work ourselves.

The trick to solving this problem is to use some undocumented features that are provided by Sun in its JDK. Taking a peek inside the classes.zip file from the Sun JDK (using any of the various zipfile utilities), we find not only the standard Java packages such as java.applet but also sun.audio. (These are in the directory sun/audio.)

The sun.audio package contains everything we need to be able to play audio clips! Here's the code:

```
import sun.audio.*; //import the sun.audio package
import java.io.*;
/** add this into your application code as appropriate
// Open an input stream to the audio file.
InputStream in = new FileInputStream(filename);
// Create an AudioStream object from the input stream.
AudioStream as = new AudioStream(in);
// Use the static class member "player" from class AudioPlayer
// clip.
AudioPlayer.player.start(as);
// Similarly, to stop the audio.
AudioPlayer.player.stop(as);
```

### JW's Most Read

#### Top 5

- How the Oracle-Google trial could affect developers
- Let's stop the FUD about JavaFX and deprecate the Java Plug-in
- The ugly underbelly of coder culture
- RestTemplate with basic authentication in Spring
- 4 good reasons to jump to HTML5

## Application Performance Management

APP SERVERS, SERVERS, DATABASES, CLOUD SERVICES, SAP, VIRTUAL SERVERS, etc ..

ManageEngine Applications Manager

DOWNLOAD FREE TRIAL

### Featured White Papers

Web app "Down for Maintenance"? Never again!

### Newsletter sign-up [View all new letters](#)

#### Enterprise Java Newsletter

Stay up to date on the latest tutorials and Java community news posted on JavaWorld

Subscribe

### Sponsored Links

- Career opportunities at Harman- Careers That'll Rock Your Socks Off. [Learn More!](#)
- Developer-friendly Java PDF libraries by Qoppa Software [Live Demos](#)
- Stop blaming your app! Diagnose Java performance issues. [Free tool >](#)
- UrbanCode: Go From Continuous Integration to Continuous Delivery [Learn More!](#)

### Sponsored Links

**Optimize with a SATA RAID Storage Solution**  
Range of capacities as low as \$1250 per TB.  
Ideal if you currently rely on servers/disks/JBODs

To use a URL as the audio stream source, substitute the following for the input stream and audio stream setup:

```
AudioStream as = new AudioStream (url.openStream());
```

Playing the audio stream continuously adds a bit more complexity:

```
// Create audio stream as discussed previously.
// Create AudioData source.
AudioData data = as.getData();
// Create ContinuousAudioDataStream.
ContinuousAudioDataStream cas = new ContinuousAudioDataStream (data);
// Play audio.
AudioPlayer.player.play (cas);
// Similarly, to stop the audio.
AudioPlayer.player.stop (cas);
```

And there you have it. Remember, this technique uses undocumented features; there are no guarantees that it will work with anything but the current Sun JDK.

#### About the author

Chong Ser Wah is a consultant at the Competency Centre for Java in Singapore. Check out the center's Java Cup Competition.

[Print](#) [Feedback](#)

Like 1 person liked this.

Comments powered by

Our Commenting [Network](#) | [Policies](#) | [Privacy](#)

#### Add New Comment

[Login](#)




Showing 20 of 22 comments

Sort by newest first ▾



**Simon Arriola**

hi folks, just want to share this simple beat maker. it ain't pretty but it shows you how to do it

```
/* * To change this template, choose Tools | Templates * and open the template in the
editor. */package beatmaker;/** * @author simon */import
java.awt.event.KeyEvent;import java.awt.event.KeyListener;import sun.audio.*;import
java.io.*;import java.util.logging.Level;import java.util.logging.Logger;public class
mainFrame extends javax.swing.JFrame {    InputStream in1;  InputStream in2;
InputStream in3;  InputStream in4;  InputStream in5;  InputStream in6;
InputStream in7;  InputStream in8;    AudioStream as1;  AudioStream as2;
AudioStream as3;  AudioStream as4;  AudioStream as5;  AudioStream as6;
AudioStream as7;  AudioStream as8;  /** * Creates new form mainFrame */
public mainFrame() throws FileNotFoundException, IOException {
this.setFocusable(true);          initComponents();    initKeyListener();
}  /** * This method is called from within the constructor to initialize the form. *
WARNING: Do NOT modify this code. The content of this method is always *
regenerated by the Form Editor. */ @SuppressWarnings("unchecked") //
<editor-fold defaultstate="collapsed" desc="Generated Code">    private void
initComponents() {    ...
```

[show more](#)

04/05/2012 11:40 PM 🚩

Like Reply



**Kevin Murphy**

Default constructor cannot handle exception type FileNotFoundException thrown by implicit super constructor. Must define an explicit constructor.

02/06/2012 09:03 AM 🚩

Like Reply



**Faraz Hashmi**

I have tried to play the sound (\*.mp3)  
 But I am getting exception error :  
 java.io.IOException: could not create audio stream from input stream  
 Please Reply...

12/04/2011 04:41 AM 1 Like

Like Reply

**SergeantBalthazar**

You can only do this with \*.wav format. To play .mp3 files you entirely different code.

03/16/2012 08:56 PM in reply to Faraz Hashmi

Like Reply

**Anonymous**

can you please send me a code that plays a audio file in jdk...  
 can you please send me a code that plays a audio file in jdk.

asheshrocks@gmail.com

Thanks in advance

07/03/2010 10:34 PM 1 Like

Like Reply

**Anonymous**

Programming  
 Object Oriented programming in java

04/19/2010 02:44 AM 1 Like

Like Reply

**Anonymous**

Alternative: convert files to native java code  
 There is an alternative to importing the sound files which works in both applets and applications: convert the audio files into .java files and simply use them in your code.

I have developed a tool which makes this process a lot easier. It simplifies the Java Sound API quite a bit.

<http://stephengware.com/projec...>

Hope this helps.  
 -- Stephen

01/17/2010 08:29 PM 1 Like

Like Reply

**Anonymous**

RE: audiodata not working  
 possibly because the format ur trying to load is not a .wav...which is a stream of sound.

12/04/2009 08:36 PM

Like Reply

**Anonymous**

javascript  
 shuold bring animation on the screen from time to time as specified by the user.

11/24/2009 08:46 PM 1 Like

Like Reply

**Anonymous**

Another more standart way  
 Although you can't use AudioClip you can use the javax.sound.sampled.Clip interface:

```
Clip clickClip = AudioSystem.getClip();
AudioInputStream ais = AudioSystem.getAudioInputStream(waveFile);
clickClip.open(ais);
clickClip.loop(times - 1);
clickClip.start();
```

11/11/2009 10:12 PM 2 Likes

Like Reply

**Anonymous**

hai

10/26/2009 07:00 AM 1 Like

Like Reply

**Anonymous**

audiodata not working  
how come the as.getData() throws an exception

10/04/2009 10:34 AM 1 Like

Like Reply

**Anonymous**

Get data limits the file to one mega byte, kinda lame

07/27/2010 05:28 PM in reply to Anonymous 1 Like

Like Reply

**Anonymous**

not working

09/15/2009 09:48 AM 1 Like

Like Reply

**Anonymous**

my solution  
u execute a java audio program without need any url  
i give a code bellow but it will execute .au, .aif, .wav files only  
\*\*\*\*veerendra\*\*\*\*\*

```

program:
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.sound.sampled.*;

public class AudioPlayer02 extends JFrame
{

    AudioFormat audioFormat;
    AudioInputStream audioInputStream;
    SourceDataLine sourceDataLine;
    boolean stopPlayback = false;
    final JButton stopBtn = new JButton("Stop");
    final JButton playBtn = new JButton("Play");
    final JTextField textField = new JTextField("Track1.au");

    public static void main(String args[])
    {
        new AudioPlayer02();
    }
    //-----//

    public AudioPlayer02() //constructor
    {

        stopBtn.setEnabled(false);
        playBtn.setEnabled(true);

        //Instantiate and register action listeners
        // on the Play and Stop buttons.
        playBtn.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                stopBtn.setEnabled(true);
                playBtn.setEnabled(false);
                playAudio();//Play the file
            }
        });
        //end ActionListener
        //end addActionListener()

        stopBtn.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e)
            {
                //Terminate playback before EOF
                stopPlayback = true;
            }
        });
        //end ActionListener
        //end addActionListener()

        getContentPane().add(playBtn,"West");
        getContentPane().add(stopBtn,"East");
    }
}

```

```

getContentPane().add(textField,"North");

setTitle("Creative Minds");
setDefaultCloseOperation(EXIT_ON_CLOSE);
setSize(250,70);
setVisible(true);
} //end constructor
//-----//

//This method plays back audio data from an
// audio file whose name is specified in the
// text field.
private void playAudio()
{
    try
    {
        File soundFile = new File(textField.getText());
        audioInputStream =...

```

[show more](#)

07/03/2010 10:25 AM in reply to Anonymous 3 Likes

[Like](#) [Reply](#)**Anonymous**

where i put the sound file  
in java, where i put the sound file?? if i want to call, how??

09/12/2009 08:28 AM

[Like](#) [Reply](#)**Anonymous**

Sorry, but...  
Applet.newAudioClip(URL) does work, I tried it in an application:

```

try {
    AudioClip ac = Applet.newAudioClip(new File(".\\sound.au").toURI().toURL());
    ac.play();
}
catch(MalformedURLException e) {}

```

java.net.URI, java.net.URL and java.io.File are all required.

You can also use AudioClip.loop() aswell.

Thomas

09/04/2009 08:45 AM 1 Like

[Like](#) [Reply](#)**Anonymous**

only plays sound once

when you start up the program again the sound goes away.

07/20/2009 11:24 PM 1 Like

[Like](#) [Reply](#)**Anonymous**

you forgot Applet.newAudioClip(URL)?  
newAudioClip(URL) is a static method in Applet Class, so it can easily be used in Applications as well.

e.g.

```

AudioClip ac = Applet.newAudioClip(url);
ac.play(); //play once
ac.stop(); //stop playing
ac.loop(); //play continuously

```

Here url can refer to some file within the jar file. So, this method is very useful.

07/13/2009 11:54 AM 1 Like

[Like](#) [Reply](#)**Anonymous**

Java  
DOES NOT WORK!

06/03/2009 01:08 PM 1 Like

[Like](#) [Reply](#)[✉ Subscribe by email](#) [RSS](#)[Load more comments](#)

Resources

TECH BRIEFCASE

Get your research on

Search, store, and share IT white papers from across the web.

Try Tech Briefcase Now

No registration required

Sponsored Links

[Java Development Outsourcing: nearshore agile Java development.](#)  
[Java Web Hosting | DailyRazor.com - Over 10yrs Of Java Web Hosting Exp!](#)  
[Java out of memory error? Use heaps up to 512GB without GC pauses. Click here for pauseless GC white paper.](#)  
[Kendo UI is everything you need to build HTML5 sites and mobile apps](#)  
[File APIs for Programmers: DOC, XLS, PPT, PDF & more. Try a free trial.](#)  
[Jelastic - next gen Java PaaS, No code changes required, Try FREE now!](#)  
[Managing Apps in the Cloud is a breeze with New Relic. See for yourself.](#)  
[Web hosting reviews for you to choose a better host](#)  
[ManageEngine: End-to-End Java Performance Management, Download Product Now!](#)

RESEARCH CENTERS	IDG Network			
Core Java	CFOworld	GamePro	IDG Ventures	Macworld
Enterprise Java	CIO	Games.net	InfoWorld	Network World
Mobile Java	Computerworld	IDG Connect	ITworld	PC World
Tools & Methods	CSO	IDG Knowledge Hub	JavaWorld	
JavaWorld Archives	DEMO	IDG TechNetwork	LinuxWorld	

About Us | Advertise | Contact Us | Terms of Service/Privacy | AdChoices

Copyright, 2006-2012 Infoworld, Inc. All rights reserved.