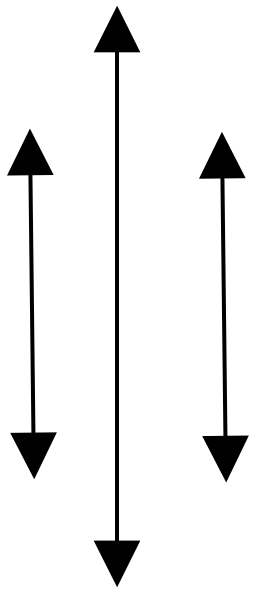
TRIBHUVAN UNIVERSITY

Institute of Engineering
Pulchowk Campus
Department of Electronics and Computer Engineering



A Final project Report
ON Minor Project

"Java Media Player"

Submitted By

Bisharjan Pokharel(061bct512) E-Mail: howareyoubishu@hotmail.com ph:-9841702007
Prakash Poudel(061bct531) E-Mail: rajpoudel@hotmail.com ph:-9841812730

Submitted To

Department of Electronics and Computer Engineering(IOE)

Date Of Submission:- 2008-2-

Acknowledgement:-

We are greatly thankful to the all who inspired us to do this project.

Especially we would like to thank **Sir Jaya Ram Timilsina**, **Sir Bikash Shrestha**, and **Sir Deepen Chapagain** for providing us a lot of help related to our project.

At last we thank all our **friends** and the **seniors** for providing necessary help and materials for the purpose of our project.

Bisharjan Pokharel (061bct512)
Prakash Poudel (061bct531)

ABSTRACT: -

"Java Media Player" is a project, which is of great use at the present. The main goal of this project is to perform various tasks, which can be done on media. **"Java Media Player"** specially focuses on playing different types of audio and video format. Capturing audio and video data and saving it to a file is done with the help of **"Java Media Player".** We can transfer and receive real time media, which makes this project valuable for the purpose such as distant education with the enhancement of this project. Similarly changing the format of the file is another purpose of this project. **"Java Media Player"** is currently focused on changing the file from WAV file format to MP3 file format that can be extended to change the file format from one format to the different formats.

For the completion of this project we use the API called Java Media Framework. It is a good API, which is used to perform the tasks, related with the media. We have used Windows as the platform for the completion of this project. Different class and package of Java Media Framework are used to achieve the task. Different types of the Manager are present in Java Media Frame work, which have different functions to perform with.

Table Of Content

Introduction

“Java Media Player” is a complete media package. It holds the entire basic requirement for playing various audio and video files with an excellent GUI. It is mainly based on the JMF API of java. JMF is a good API for works related to the media. **“Java Media Player”** can be used for capturing audio and video using microphone and webcam. With the further improvement it also can be used as a tool for voip. **“Java Media Player” can** be used for the converting one file format to another format. This version of **“Java Media Player”** is capable of converting the WAV file to MP3 file format, which can be extended to support other file format. Similarly it can be used to transmit media files across network and receive those files from the network. All the transmitting and receiving media files is based on UDP protocol. UDP protocol is mainly used for the purpose of multimedia file transfer. Since all transmitting and receiving is done on real time protocol **“Java Media Player”** can also be extended for works such as videoconference and distant education. Thus **“Java Media Player”** is software with many useful features at the present world.

OBJECTIVES: -

The main objectives of the **"Java Media Player" can** be listed as follow:-

- ✓ Playing audio files
- ✓ Playing video files
- ✓ Capturing audio from audio devices
- ✓ Capturing video from video devices
- ✓ Transmitting real time Media across the network.
- ✓ Receiving real time Media
- ✓ Converting Media from one format to other format
- ✓ To provide basic functions such as pause, stop, play, next and so on in a media player
- ✓ To maintain a play list.

Project Background :-

For the completion of **"Java Media Player"** a high level programming language JAVA is used and windows platform is used. But the result of this project is platform independent. We have used the latest version of the Java programming language for the latest feature included in java such as tray and spleen screen. Similarly capture device is required for the completion of the project. We have a team of two men for the completion of the project. But help from seniors, teachers and friends are most for the completion of this project.

The main propose of this project is to perform the tasks related to the media, such as playing audio and video, capturing the medial, transmitting and receiving the media, changing the format of the media from one to another .The main source for the completion of this project is the web, www.sun.com/java, Java media framework API, Java How To Program by Deitel and other articles related to the media from the web.

This project can be further enhanced in future to the mobile application and even be expanded to contain many more facilities so that it can be very useful.

Description Scenarios

Scenario 1

If a person wants to listen music or watch video on computer then he/she can use this **"Java Media Player."** He/she can play one or more file, which appears on play list. While listening some files the person can add more files to the play list using the add bottom. One can adjust the size of the screen on which the video is displayed. **"Java Media Player"** has a function of keeping to the tray and one can view the smaller size of the video in the tray of the system. One can keep or throw the play list in the video screen. This can be done just by the help of the button and to show again the play list one can click the show play list in menu .The files in the play list can be randomized. One can repeat whole songs or can continuously play the same file-using repeat all and repeat this item in the menu respectively.

Scenario 2

"Java Media Player" is useful for the people to record video and audio. These recorded audio and video can be stored to a file, which can be used in future. Different programs contains different sounds which can be generated with the help of **"Java Media Player"**

The help of recording capability of it can also convert the recorded files converted to different formats from one format.

Scenario 3

If a person on a network wants to listen to a audio file or watch a video on the server he can just use **"Java Media Player"** for this purpose. For this purpose server contains the files that can be requested by the person who wants to listen/watch and the server sends it across the network. This it can be used for the purpose of broadcasting, distant education, and so on.

Case study

We have studied on various web articles related to the media player and streaming he media. Particularly we have studied about different articles on the web related to the JMF API of java. We have studied and searched different articles and documentation related to different media players. We have studied about different plugins related about different media formats. We have researched on the topics such as streaming and transmitting media across networks. A lot of research has been done on changing the formats of the media file.

Jigul 3.0 is an open source media player, which we found on the net on which we spent time for the research. We tried to implement the sound effect with the help of Jigul 3.0.

Similarly we have studied documentation of a media player called gz player.

System's Block Diagram:-

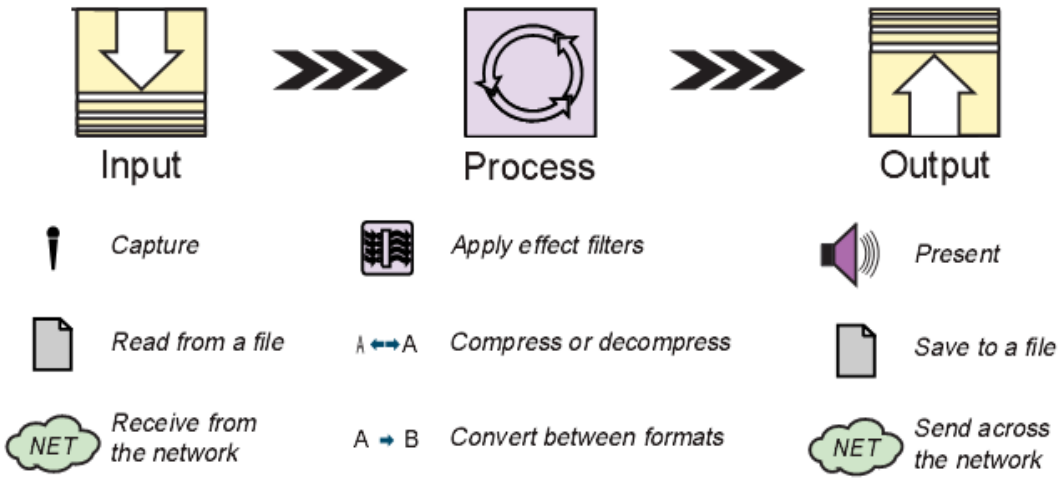


Figure 1-1: Media processing model.

Methodology: -

Different steps were implemented during the preparation of the projects.

The following methods were employed for **"Java Media Player":**

- ✓ **Sample survey**:- In this a brief research was made related on the project.
- ✓ **Information**:- Required information was collected from the different source such as web and different sources.
- ✓ **Opinion collection**:- Opinion related to the project was collected from different source such as friends and teachers.
- ✓ **In depth study**: - finally in depth study was made related to the project. Different way of implementing the found resource was studied.
- ✓ **Programming**: - After all of the above programming is done.
- ✓ **Testing and maintanence**: - after programming testing and maintenance was done.

The Project Schedule:-

The schedule of the project is as follow.

The estimation of the time is as follow: -

ID	Task	Duration(days)	Commutative days
1)	Choose project	7	7
2)	Discussion	20	27
3)	Resource collection	10	37
4)	Proposal writing	3	40
5)	Programming preparation	20	60
6)	**Coding**		
6.1)	Playing audio	2	62
6.2)	Playing video	5	67
6.3)	Capturing media	10	77
6.4)	Transmitting media	20	97
6.5)	Receiving media	20	117
6.6)	Changing format	10	127
6.7)	Creating GUI	10	137
7)	Debugging	10	147
8)	Report writing	3	150

Literature Review

Java Media Framework

The Java[a] Media Framework (JMF) is an application-programming interface (API) for incorporating time-based media into Java applications and applets. This guide is intended for Java programmers who want to incorporate time-based media into their applications and for technology providers who are interested in extending JMF and providing JMF plug-ins to support additional media types and perform custom processing and rendering.

Time based media

Any data that changes meaningfully with respect to time can be characterized as time-based media. Audio clips, movie clips, and animations are common forms of time-based media. Such media data can be obtained from a variety of sources, such as local or network files, cameras, microphones, and live broadcasts.

Content Type

The format in which the media data is stored is referred to as its *content type.* QuickTime, MPEG, and WAV are all examples of content types. Content type is essentially synonymous with file type - content type is used because media data is often acquired from sources other than local files.

Media Streams

A media stream is the media data obtained from a local file, acquired over the network, or captured from a camera or microphone. Media streams often contain multiple channels of data called tracks. For example, a QuickTime file might contain both an audio track and a video track. Media streams that contain multiple tracks are often referred to as multiplexed or complex media streams.

Media Processing

In most instances, the data in a media stream is manipulated before it is presented to the user. Generally, a series of processing operations occur before presentation:

1. If the stream is multiplexed, the individual tracks are extracted.
2. If the individual tracks are compressed, they are decoded.
3. If necessary, the tracks are converted to a different format.
4. Effect filters are applied to the decoded tracks (if desired).

Media Capture

Time-based media can be captured from a live source for processing and playback. For example, audio can be captured from a microphone or a video capture card can be used to obtain video from a camera. Capturing can be thought of as the input phase of the standard media-processing model. A capture device might deliver multiple media streams. For example, a video camera might deliver both audio and video. These streams might be captured and manipulated separately or combined into a single, multiplexed.

Stream contains both an audio track and a video track.

Managers

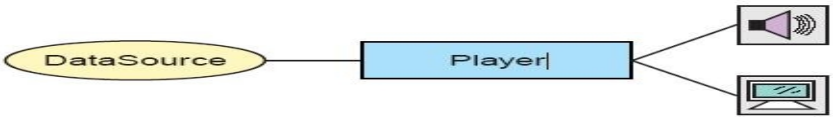
The JMF API consists mainly of interfaces that define the behavior and interaction of objects used to capture, process, and present time-based media. Implementations of these interfaces operate within the structure of the framework. By using intermediary objects called managers.

JMF uses four managers:

1. **Manager**: -handles the construction of Players, Processors, Data Sources, and Data Sinks. This level of indirection allows new implementations to be integrated seamlessly with JMF. From the client perspective, these objects are always created the same way whether the requested object is constructed from a default implementation or a Custom 1.
2. **PackageManager**-maintains a registry of packages that contain JMF classes, such as custom Players, Processors, Data Sources, and Data Sinks.
3. **CaptureDeviceManager**: -maintains a registry of available capture devices.
4. **PlugInManager**:-maintains a registry of available JMF plug-in processing components, such as Multiplexers, Demultiplexers, Codecs, Effects, and Renderers.

Players

A Player processes an input stream of media data and renders it at a precise time. A Data Source is used to deliver the input media-stream to the Player. The rendering destination depends on the type of media being presented.



Player

States

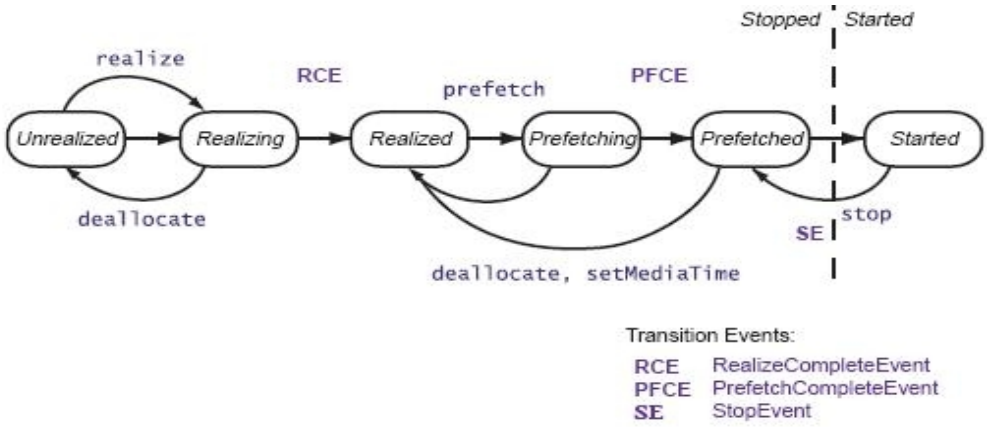
A Player

can be in one of six states. The Clock interface defines the two

Primary states: Stopped and Started. To facilitate resource management,

Controller breaks the Stopped state down into five standby states: Unrealized,

Realizing, Realized, Prefetching, and Prefetched.



Processors

Processors can also be used to present media data. A Processor is just a specialized type of Player that provides control over what processing is performed on the input media stream. A Processor supports all of the same presentation controls as a Player.

Streaming Media

When media content is streamed to a client in real-time, the client can begin to play the stream without having to wait for the complete stream to download. In fact, the stream might not even have a predefined duration - downloading the entire stream before playing it would be impossible. The term streaming media is often used to refer to both this technique of delivering content over the in real-time and the real-time media content that is delivered.

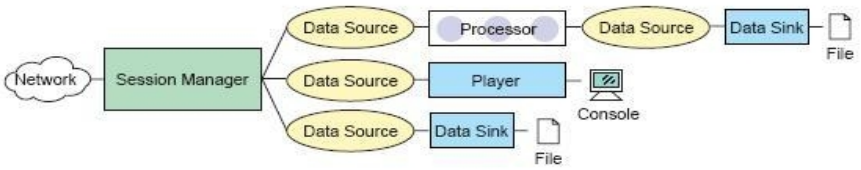
Protocols for Streaming Media

Transmitting media data across the net in real-time requires high network throughput. It is easier to compensate for lost data than to compensate for large delays in receiving the data. This is very different from accessing static data such as a file, where the most important thing is that all of the data arrive at its destination. Consequently, the protocols used for static data don't work well for streaming media. The HTTP and FTP protocols are based on the Transmission Control Protocol (TCP). TCP is a transport-layer protocol1 designed for reliable data communications on low-bandwidth, high-error-rate networks. When a packet is lost or corrupted, it is retransmitted. The overhead of guaranteeing reliable data transfer slows the overall transmission rate. For this reason, underlying protocols other than TCP are typically used for this reason, underlying protocols other than TCP are typically used for streaming media. One that is commonly used is the User Datagram Protocol (UDP). UDP is an unreliable protocol; it does not guarantee that each packet will reach its destination. There is also no guarantee that the packets will arrive in the order that they were sent. The receiver has to be able to compensate for lost data, duplicate packets, and packets that arrive out of order. Like TCP, UDP is a general transport-layer protocol-a lower-level networking protocol on top of which more application-specific protocols are built. The Internet standard for transporting real-time data such as audio and video is the Real-Time Transport Protocol (RTP).

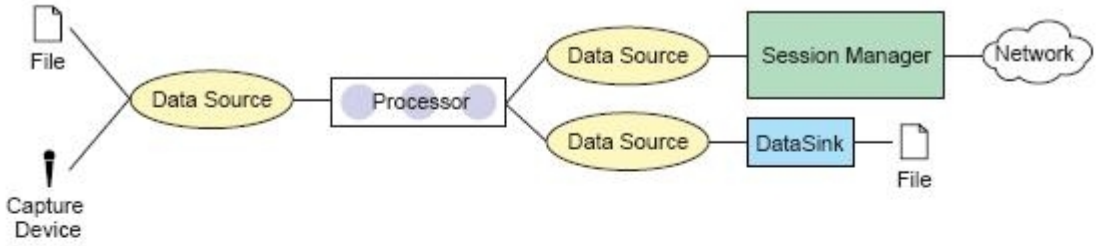
Real-Time Transport Protocol

RTP provides end-to-end network delivery services for the transmission of real-time data. RTP is network and transport-protocol independent, though it is often used over UDP.

RTP Reception: -



RTP Transmission: -



System Tray and Splash Screen

System Tray and Splash Screen is the new feature supported by java 6 to improve GUI.

System Tray

It is a specific area, which can be accessed during running program. It appears at the bottom of the desktop. Any user can access it continually. All running application on the desktop can share this tray area. It is referred to as the Taskbar Status Area on the Microsoft windows, the system Tray on KDE, and the Notification Area on the Gnome Desktop. This system Tray can be accessed with two separate classes defined in Java.awt.SystemTray;

Java.awt.TrayIcon;

Splash Screen

Splash Screen informs user that the applications is starting up. It is a standard part of any GUI application. A polished Splash Screen of the application can be used for presenting marketing information and legal information, such as copyright information, third party logo and so on.

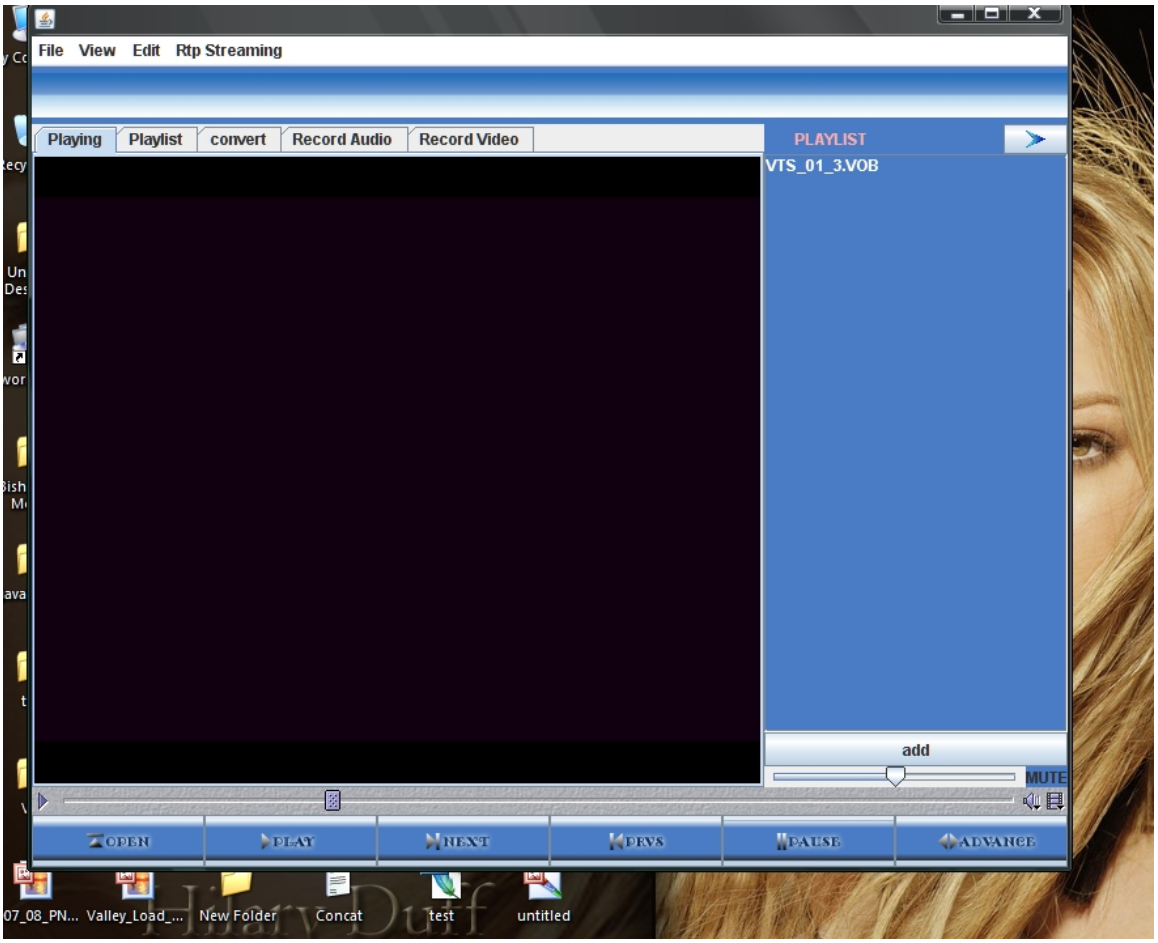
For creating the Splash Screen in Java, we can use AWT/Swing. There should be a minimum delay between the application's startup and the pops up of the splash screen. The Java SE version 6 provides a good feature to show the Splash Screen. It allows the application to show the Splash Screen much earlier, even before the Java Virtual Machine starts. Now, it is possible to decode the image and then display it in a simple window by an application launcher. The Splash Screen is also capable for displaying any image format like jpeg, gif, png with transparency and animation. For Splash Screen package used is

Java.awt.SplashScreen;

Scope: -

Scope of “Java Media Player” is simple Personal Computer to a large server. “Java Media Player” can be used for playing audio and video files and for transmitting the real time media to large distance. It can be use to capture audio sound via microphone and video from devices like webcam. So “Java Media Player” can be a very useful project for the distant education and for transmitting the live programs.

Screen Shot:


File View Edit Rtp Streaming
Playing
Playlist
convert
Record Audio
Record Video
PLAYLIST
VTS_01_3.VOB
add
MUTE
OPEN
PLAY
NEXT
PREVS
PAUSE
ADVANCE
07_08_PN...
Valley_Load_...
New Folder
Concat
test
untitled
Hilary Duff

Future Enhancement:

There are certain limitations in **"Java Media Player."** Adding and implementing some additional features can remove these limitations. These features include:

- ✓ Java Media Player can be further enhanced by adding equalizer for having effects on the sound
- ✓ Visualization can be added as future enhancement in Java Media Player
- ✓ As Java Media Player has the feature like capturing audio/video and rtp streaming which can be used to make VoIP.

Conclusion:-

We have successfully completed our project. While doing our project we faced different difficulties. Those difficulties are actually removed with the help of our seniors, colleagues, and other different resources. After doing this project we have learnt visual programming viz. Java and also gained the idea of socket programming with various Internet standard used in Java Media Framework. The more important thing is that we have learnt to work in-group. As well we have learnt the importance of the project management while doing any sort of project.

References:-

1) **www.sun.com/java**
2) **JMF API guide**
3) **Different java forums**
4) **Many web articles**
5) **How to program. By Deitel and Deitel**
6) **Java Programming Black Book**