

Preface

Over the years Java has become a stable, expressive, and secure language for Internet programming. The success of the Java language can be attributed to the following three features, (i) the sound design of the basic language, (ii) categorization of the three platforms namely standard, enterprise and micro edition each one tuned to an usage scenario, and (iii) the rich collection of APIs (basic and optional) each one providing a high level abstraction in an application domain.

Internet is becoming more and more pervasive. Also a sizable volume of the Internet traffic is multimedia in nature. It is then natural that we expect a Java API for multimedia application. Sun Microsystems and its industry partners have realized the need for such an API as early as 1996 and introduced the Java Media Framework 1.0 (JMF). Over the years JMF has gained popularity and has proved to be a powerful API for developing multimedia applications in Java.

Few books have been written on JMF. To the best of the author's knowledge as on date there is no book covering extensively all the aspects of JMF 2.1.1, the latest version of JMF. The present book is aimed at filling this void.

My own experience with JMF started and grew by guiding a number of student projects both at the bachelor's and at masters level at the EE department of IIT Madras. The early days were simply fascinating. The power and the fun of handling audio and video the way you want through your Java program hooked us onto the JMF.

The serious work with JMF started when I decided to develop the Cosmophone application by extending the ideas used in a couple of student projects. The JMF API specification and the JMF user's guide were very helpful during the project work. However much has to be learnt through experience. Lack of books has made a typical JMF programmer to spend considerable amount of time before becoming productive. This is yet another reason that prompted me to write this book.

The book has been divided into four parts. Anyone who would like to have a feel for JMF and a basic working knowledge in JMF can just read part-I alone. Knowledge in multimedia is not a pre-requisite to read this book. Chapter 3 on multimedia makes the book self-contained. Chapter 4 " A quick tour to JMF" gives a larger picture to JMF. You may like to go-through this chapter a second time after finishing part-I, to assimilate part-I and as an orientation for part-II. Sections 5.4, 5.5, 5.6, and 6.8 can be omitted during the first reading of the Part-I.

If you are only interested in the playback of multimedia contents Chapters 5 and 6 will suffice. Chapter 7 tells you how to capture media, while chapters 8 and 9 illustrates all about media processing. Chapter 10 explains the technique of storing media in a file. Simple media transmission and reception can be achieved using chapter 11. The chapter on JMF applet will be of interest to those planning to develop applets with multimedia capabilities.

Part-II deals with advanced topics of JMF. If you are interested in developing multimedia applications involving exchange of multiple media streams(such as audio and video), or exchange of media between more than two distributed programs (for example video-conferencing application) you should better use JMF RTP API. The JMF RTP API and its power of handling complex multimedia exchange in an application is explained in chapters 13, 14 and 15. To have a fine-grained control over your JMF applications primarily to improve

its performance read chapter 16 on JMF Control. Chapter 17 aids in developing large-scale and complex applications involving multiple media players and sources.

Software engineering principles encourage use of software components in software development. In the world of Java the standard for components are JavaBeans. JMF supports development of multimedia bean and those concepts are explained in chapter 18. One of the nice features of JMF is that you can extend JMF to support additional media formats, media access protocol etc. You can also implement your own media players or processors to have better performance. We study these aspects in chapter 19.

Part-III of this book is a rather detailed case study of a multimedia application called cosmophone. Cosmophone is a four-in one application having (i) video telephony (ii) video mailing (iii) video conferencing application and (iv) answering machine. The reason why we discuss cosmophone application in detail is because it touches most of the aspects of JMF presented in Part-I & II.

Certain topics of special interest in JMF are discussed briefly in the last part of this book. These include the MMAPI, IBM's HotMedia, JMF Customizer and the anatomy of the JMF. The topic on Mobile Media API (MMAPI) covers the multimedia package for the J2ME platform. The MMAPI framework closely resembles the JMF. In this topic we explain the MMAPI with a number of example multimedia MIDlets. HotMedia is a Java based tool for authoring web pages that are rich in multimedia contents. The JMF Customizer is a tool for the developers to customize JMF for the purpose of packaging the JMF along with their multimedia applications. The anatomy of JMF gives an insight to the structure of the JMF reference implementation.

Attempt has been made to cover or at least touch upon most the aspects of JMF (which has resulted in a rather large gestation period of one and a-half-years for this book !). All the programs presented in this book have been developed and tested on the JMF version 2.1.1 Windows performance pack.

Before you go through this book here is a word of caution. Do not consider this book as an authoritative solution to your problems in JMF. I am not an expert in JMF. Experts are the engineers at the Sun Microsystems who have designed and developed the JMF. I am just a user of JMF who happen to have worked with JMF for about three years. My knowledge of JMF is through the literature published by Sun on JMF. What I present here is the JMF as perceived by me. I apologize for the conceptual, or typographical mistakes or mistakes related to legal aspects that might have crept in this book. I might have misinterpreted, overlooked, or de-emphasized certain aspects of JMF. I welcome your criticisms for improving the future editions of this book.

T.G.Venkatesh