Dedication

This book is lovingly dedicated to Samantha, David, and Irene. Thanks for everything, you're the best.

Acknowledgments

Jeff Eisen provided lots of technical review and assistance with the manuscript. Whatta guy. Thanks also to Bob Congdon for help with and advice on the servlet examples. No list of thank yous could be complete without acknowledging the years of support and assistance and inspiration I received at Iris Associates from Len Kawell and Tim Halvorsen. And thanks to Lance and Willie for being their respective selves.

Preface

Why Java? Why Domino? Why This Book?

Are all authors (especially first-time authors) so defensive that they feel they have to explain why they're writing a book? Perhaps not. Still, I can't assume that the gazillion people out there who work at developing killer collaborative/groupware/client-server/Internet/intranet applications understand why they should care about Lotus Domino, or Java, or both.

After ten and a quarter years (ten years, three months and six days, to be precise) of writing software for Lotus Development Corp. and its subsidiary Iris Associates, I left to work on other projects, this book included. While I was at Lotus/Iris, I spent about four and a half years working on Notes, which was later renamed Domino. I watched the product get reinvented a couple of times on its way to becoming what it is now—a terrific tool for building collaborative and group aware applications—although the journey is not yet finished.

Lots of people now know how good Domino is. Lots of people (not always the same ones) also now know how good the Internet is at facilitating communication among computers. Lots of people have also figured out that Java is a great tool for building all kinds of Internet-aware applications. I figured that it was time more people became aware that Domino and Java are an amazingly powerful combination.

The point of this book is to show you how to use the Java programming language to manipulate Domino objects. I'll leave it to you to decide how best to use those Domino objects to create your next killer Internet app. I'll stick to showing you how to use the features of Java and Domino to best advantage.

Is Domino 4.6 (the first release of the product that includes a Java API) perfect? By no means. As with most software products, it evolves and improves over time. In some respects, the Java features in Domino 4.6 represent baby steps, to be improved upon in future releases. You decide.

Much of this book is about something called the Notes Object Interface, or NOI. No matter how great a development platform Domino/Notes is, it won't *really* support robust, production-quality applications unless it's programmable. Intuitive user interfaces, wizards, and all that are great to have, but there are times when you need to really get down and tell the product *how* to do something. That's why Notes has always had a formula language, why LotusScript was added in release 4.0, and why a Java interface was added in release 4.6.

Both the LotusScript and Java interfaces to Notes expose a set of *objects* particular to the Domino/Notes product. Each object type is a *class*, and each class exposes certain attributes (*properties*) and behaviors (*methods*). The classes are things like Database, Document, View, Agent, and so on—all the objects with which you may have already become familiar by using the product. Together, these classes make up the Notes Object Interface.

Most of what you do when you use Java to program Domino is write code that manipulates NOI objects in some way. This book shows you how to do that. It goes beyond the excellent on-line documentation provided with the product to give you more in depth discussion and analysis of how the classes work together and why they are the way they are. I've put together lots of examples, all of which are discussed in some detail. All of the source code for the examples, and all the Notes databases that go with them, are available on the CD-ROM you'll find at the back of the book.

Who should read this book? I think two groups of people would benefit most:

1. Notes/Domino application developers who already know LotusScript and want to learn Java to see how it can help them create better Inter/intranet apps.
2. Java programmers who want to learn how to use the Notes Object Interface.

This book is not meant to teach you the Java programming language if you don't already know it. It is meant to teach you how to use Java to the fullest advantage when programming NOI. What we're striving for here is to give you a richness and depth of understanding that you won't get from the Domino on-line documentation, with lots of examples plus maybe a few surprises. I'll point out a few places where the Domino 4.6 documentation is incorrect (there aren't very many).

Chapter Overview

Chapter 1, "Domino/Notes Programmability Overview," is a survey of the commonly used Internet programming languages and how they are, or might be, used in the context of a Domino server and/or a Notes Client. It attempts to set the stage for the following chapters by giving you an overview of the Notes Object Interface (NOI) and scripting (both client side and server side).

Chapters 2–6 provide an in-depth look at the individual classes that make up the NOI. Each class's Java binding is discussed in some detail, and examples are given. All code examples (and any of the Notes databases used in the examples) are provided on the CD-ROM found at the back of this book, as well as on my Web site (http://www.looseleaf.net).

Chapter 7 is all about writing standalone applications using Java and NOI. It goes into some detail on how to write both single and multithreaded programs.

Chapter 8 tells you how to code Java Agents for Domino, both single and multithreaded. Chapter 9 shows you how to use third-party Java development tools to debug Java Agents.

Chapter 10 goes into more detail on multithreaded Java programs using NOI, and discusses how (and how not) to share Notes objects across threads.

Chapter 11 tells you about servlets and how to write them (in case you are using an HTTP server other than Domino, such as Lotus Go). It also tells you how to convert your servlets into Agents, for when you upgrade to Domino. Again, all the code and databases used in the examples are on the CD.

Chapter 12 discusses Java Beans and their relationship to NOI.

Chapter 13 shows you how to conveniently (and profitably) use JDBC in combination with NOI to write Java programs that mix and match Notes and relational DBMS access.

Finally, Chapter 14 gives you a sneak preview of some upcoming Domino technologies, and my personal theories on where some of this stuff is headed in future releases of Domino/Notes.

I'd love to hear what you think of this book, and invite you to visit my Web site at http://www.looseleaf.net. Enjoy.