

Inside JXTA: Programming P2P Using the JXTA Platform

Chapter 1 Introduction

This chapter will provide a general introduction to the topic of peer-to-peer (P2P) computing including a history on the development of the P2P paradigm, an overview of the technical problems involved in implementing P2P, a description of the intended audience for the book, and instructions on how to obtain and install the JDK, and JXTA source code.

Why Peer-to-Peer?

This first section outlines why peer-to-peer is an important step in the development of the Internet; basically, peer-to-peer enables people to find useful information in the mountains of data in the Internet. The web is exploding, and search engines (such as Google and Altavista) are struggling to keep pace with the growth, while still providing meaningful results to search queries. The answer is to enable the people with the information themselves to answer the queries; this distribution of resources and knowledge, a logical extension of the same mechanisms that help the Internet's network itself to be so robust, is the basis of peer-to-peer. Using P2P applications, people can reach out across the network to find specific resources, be they a person to engage in interactive chat, a music file, a video clip, or a document. A P2P application user may even reach across the network for a less material resource, such as computing power.

Add diagrams illustrating the difference between the old way of doing things and the peer-to-peer way of doing things. These should be basic, as the in-depth presentation of P2P concepts will be made in Chapter 2.

The History of Peer-to-Peer

This section outlines the history of P2P, starting with the Napster, followed by the Gnutella, and Freenet networks. Could potentially add SETI@Home and other distributed computing projects here, but may not want to muddy the waters, as these were not true P2P applications. The basic problems that these networks attempted to solve is presented, along with the drawbacks of their approaches, such as the reliance of Napster on a central server to provide the search capabilities, or the proprietary nature of all of their protocols. This section sets up the following section by outlining the need for a way for developers to produce P2P solutions, without designing their own protocol and libraries from the ground up.

Enter the JXTA Platform

Drawing on the problems of the pioneer P2P systems, the initial features of the JXTA Platform are presented: language neutrality, standards-based, transport independence, extensibility, etc. The JXTA Platform shouldn't be thought of as a library for peer-to-peer applications; although there are implementations of JXTA available currently for C and Java, the JXTA Platform is meant to be a definition of a common set of protocols that can be used to achieve peer-to-peer networking. Even if your application isn't going to be written in C or Java, or run over HTTP or TCP, the JXTA Platform can still be used to provide the framework for conducting peer-to-peer transactions.

What's In This Book

This section simply provides an overview of the rest of the book, highlighting important sections. Underlines that this book focuses on the Java implementation of the JXTA platform for its examples, but that the JXTA protocols are language independent.

What You'll Need To Try The Examples

Gives instructions for obtaining a Java Development Kit, CVS, and using CVS to retrieve the source code for the JXTA Platform.

Chapter 2 P2P Concepts

This chapter introduces the basic concepts of peer-to-peer computing, the problems posed by P2P, and the terminology required to understand the rest of the book. This chapter provides a technical introduction to the components required to implement a complete P2P solution.

P2P Terminology

Defines the basic P2P terminology used by JXTA, including: peers, peer groups, advertisements, messages, pipes, and identifiers.

Finding Peers

Answers the question: how does a peer find other peers on the network?

P2P Communication

Answers the question: how do peers communicate with each other once they're found each other; illustrates how message routing in P2P networks works to pass message from one peer onto others in order to distribute functionality across the network.

Chapter 3 Introducing JXTA

This chapter introduces the solutions JXTA provides to address the basic problems of peer-to-peer computing, the assumptions of the JXTA architecture, and capabilities and limitations of JXTA. The majority of the chapter will revolve around using the JXTA Shell application to familiar the user with the JXTA implementation of P2P, and allow some preliminary experimentation without requiring the reader to program anything just yet. In addition, a small introduction on XML will be required to familiarize the user with the XML messaging format used by all of the JXTA protocols.

The JXTA Advantage

The key advantages provided by the JXTA platform are presented: standards-based, interoperability, platform independence, and ubiquity. The purpose of the JXTA protocols is introduced only briefly, referring to the appropriate chapters for in depth explanation and examples. In addition, justification for the use of XML is presented.

Experimenting with JXTA: The JXTA Shell

This section will provide a thorough tutorial on using the JXTA Shell to explore the concepts of P2P introduced so far.

Obtaining the JXTA Shell

Instructions for downloading a pre-built version the JXTA Shell to use for the experiments appear in this section. Provides alternative instructions for obtaining the latest version of the shell from source control, and building the binaries.

Running the JXTA Shell

Provides details on how to configure the JXTA Shell application to connect to other peers; alternatively, provides instructions on how to run two JXTA shells on your own machine, so they can communicate with each other. This includes information on running the JXTA Shell as a JXTA router or resolver.

Using the JXTA Shell

This section walks the user through running commands within the JXTA Shell, allowing them to find other peers and peer groups, join and leave peer groups. A listing of the commands, and their meaning/operation/parameters is provided.

XML: A Brief Introduction

The user will need at least a rudimentary understanding of the format of XML documents in order to understand the protocol definitions in the following chapters. A brief introduction of the form of XML elements, and attributes is given. The introduction builds on the XML format of the peer information returned from within the JXTA Shell.

Identifiers in JXTA

This section provides in-depth information on the identifiers used within JXTA to uniquely identify peers, peer groups, pipes, and contents. The JXTA Shell is used to show the different types of identifiers, and where these identifiers occur.

Advertisements

The basic unit of messages between peers in JXTA, advertisements serve to describe and publish peer resources. This section describes the several types of advertisements: Peer, Peer Group, Pipe, Service, Content, and Endpoint advertisements.

Messages

Describes how message are formatted in order to provide an XML-based message transport.

Transport Bindings

Describes how the XML message transport is bound to either TCP or HTTP to obtain network connectivity.

Note: Each of the following chapters will follow the same basic format, first introducing the protocol and its purpose, then drilling down into the specifics of the protocol, and finally providing examples showing how to use the JXTA to implement that portion of P2P functionality. The general idea is to develop each of the portions of a final sample application in a standalone fashion, but in such a way that each chapter's example applications are self-contained and can be run stand-alone.

Chapter 4 The Peer Discovery Protocol

The Peer Discovery Protocol (PDP) provides P2P applications with a mechanism for finding other peer resources; without PDP, a P2P client would be useless, unable to find and use the resources offered by other peers. This chapter details the purpose of the PDP, the use of the PDP in P2P applications, the format of the PDP's XML messages, and guides the user through an example application that uses the PDP to discover other peers on a network.

Flow of the Peer Discovery Protocol

Provides a flowchart for the flow of message in the PDP; in addition, outlines the rules provided in the JXTA Protocol Specification that govern the transmission and routing of discovery messages.

The Discovery Query Message

Describes the Discovery Query message; the Discovery Query message is used to discover advertisements for services made available by other peers. The message itself can specify the type of advertisement to discover, as well as other parameters to govern the response from peers. Included in this section will be the code to produce a Discovery Query message using the Java version of the JXTA Platform libraries.

The Discovery Response Message

The Discovery Response Message is used by a peer to respond to a Discovery Query message from another peer. The contents of the message include the advertisements of resources available from the peer that is responding to the request. Included in this section will be the code to produce a Discovery Response message using the Java version of the JXTA Platform libraries. This example will be combined with the example from the previous section to produce a basic discovery solution.

Chapter 5 The Peer Resolver Protocol

The Peer Resolver Protocol (PRP) provides P2P applications with a generic request and response format to use when communicating with other peers. Once a peer has been discovered using the Peer Discovery Protocol, the PRP can be used to send messages to the peer for processing, and to receive messages from the peer containing the results. This chapter details the purpose of the PRP, the use of the PRP in P2P applications, the format of the PRP's XML messages, and guides the user through an example application that uses the PRP to send and receive simple messages between two peers.

Flow of the Peer Resolver Protocol

The Resolver Query Message

The Resolver Query message is used to send a request for a specific service from a peer to a member of a peer group. The contents of the message include sender credentials required to access the service, the name of the service's handler, and a query string to be processed by the handler. Included in this section will be an example to produce a Resolver Query message for a handler that will be developed in the next section.

The Resolver Response Message

The Resolver Response message is used by a peer to provide the response to a previous Resolver Query message sent by another member of a peer group. The contents of the message include the credentials of the sender, the name of the service handler that generated the response, and output generated in response to original query string submitted in the original Resolver Query message. This section will include an example that will create a simple handler, and the required code to accept an incoming Resolver Query message, map the query to the simple handler, and generate an appropriate Resolver Response message.

Chapter 6 The Peer Information Protocol

Once a peer has been discovered using the Peer Discovery Protocol, the status or capabilities of the peer may be required. The Peer Information Protocol (PIP) provides a set of messages capable of querying a peer in order to obtain status information. This chapter details the purpose of the PIP, the use of the PIP in P2P applications, the format of the PIP's XML messages, and guides the user through an example application that uses the PIP to query a peer for its status information.

Flow of the Peer Information Protocol

The Ping Message

The Ping message is used to check the status of a previously discovered peer, and obtain information about that peer. The Ping message consists of the source and target peer IDs, credentials of the sender, and, optionally, the type of ping response required. A peer can request either a full response, consisting of advertisements for all services made available by the peer, or a short response, consisting of a simple acknowledgement. Two examples will provide the code required to request the short and long status information from a peer, and then combine them into a single example that can provide both pieces of functionality.

The Peer Info Message

The Peer Info message is sent by a peer in response to a Ping message, and provides either a basic acknowledgement consisting of the peer's uptime, or a full acknowledgement containing all of the peer's service advertisements. The contents of the message include the credentials of the peer, the source and target peer IDs, the uptime, a timestamp, and a peer advertisement. This section will include an example that generates the appropriate Peer Info message in response to the Ping message sent by the example applications created in the previous section.

Chapter 7 The Peer Membership Protocol

The Peer Membership Protocol (PMP) provides peers with the ability to form and administer peer groups; using the PMP, a peer can request credentials to become part of a peer group, as well as change, or delete credentials. This chapter details the purpose of the PMP, the use of the PMP in P2P applications, the format of the PMP's XML messages, and guides the user through an example application that uses the PMP to interact with a peer group to create, edit, and delete their credentials with the peer group.

Flow of the Peer Membership Protocol

The Membership Apply Message

The Membership Apply message is sent by a peer to a peer group's membership application authenticator in order to receive permission to join a peer group. The apply message consists of the credential of the peer requesting membership, the source pipe ID, and the authenticator pipe advertisement. This section contains example code required to generate a Membership Apply message.

The Membership Join Message

The Membership Join message is sent by a peer to a peer group's membership application authenticator in order to join a peer group, after it has received permission to join the group (in the response to the Membership Apply message). The join message consists of the application credential received in the ack to the apply message, the source pipe ID, the membership pipe advertisement, and identity information. This section augments the previous example code to add the capability to generate a Membership Join message.

The Membership Ack Message

The Membership Ack message acts as an acknowledgement to both Membership Apply messages, and Membership Join messages. It is sent back to indicate either the success or the failure of the apply or join attempt, and contains the sender credentials, source pipe ID, a membership pipe advertisement, a peer group advertisement, and group credentials. The example in this section provides code to generate a Membership Ack message in response to both the Membership Apply and Membership Join message.

The Membership Renew Message

The Membership Renew message is sent to a peer group to renew a peer's access to a peer group. A Membership Ack message is generated in response to the sender with a new credential and lease if the renew is accepted. The renew message contains the source credentials, source pipe ID, and membership pipe advertisement. The previous two examples are augmented to provide the code to generate a renew message, and provide the appropriate acknowledgement response.

The Membership Cancel Message

The Membership Cancel message is sent by a peer to a peer group membership authenticator to cancel the peer's membership or application rights in a peer group. The last

example is augmented to add the functionality required to generate a cancel message, and generate the appropriate acknowledgement message.

Chapter 8 The Pipe Binding Protocol

Based on the idea of Unix pipes, pipes in JXTA are virtual channels connecting two endpoints; the pipe binding protocol allows peer group members to establish a connection to another peer, independent of the transport mechanism. This chapter details the purpose of the PBP, the use of the PBP in P2P applications, the format of the PBP's XML messages, and guides the user through an example application that uses the PBP to exchange information over a pipe with another peer.

Flow of the Pipe Binding Protocol

The Pipe Binding Query Message

The Pipe Binding Query message is sent by a peer pipe endpoint to find a pipe endpoint bound to the same pipe advertisement. The message contains the sender credentials, an optional peer ID, whether the response can come from a cache, and the pipe ID to be resolved.

The Pipe Binding Answer Message

The response to a Pipe Binding Query message, the Pipe Binding Answer message is sent back to the original peer by each peer bound to the pipe. The response includes the sender credential, the pipe ID resolved, and the peer URI where an input pipe has been created. This section provides an example of the code required to perform both the Pipe Binding Query and Answer messages.

Chapter 9 The Endpoint Routing Protocol

Due to the ad hoc nature of a P2P network, a mechanism is required to enable messages to be routed between peers. The Endpoint Routing Protocol (ERP) provides peers with a mechanism for determining a route to an endpoint, allowing the peer to use the Peer Binding Protocol to bind to the endpoint, in order to exchange information with the remote peer. This chapter details the purpose of the ERP, the use of the ERP in P2P applications, the format of the ERP's XML messages, and guides the user through an example application that uses the ERP to determine the route to another peer.

Flow of the Endpoint Routing Protocol

The Route Query Message

A peer sends a Route Query message is sent to a peer router to request route information. The query consists of the sender credential, the peer ID of the destination, and whether or not the response can be cached.

The Route Answer Message

The response to a Route Query message, consists of the sender credential, the peer ID of the destination, the peer ID of a router that knows the route to the destination peer, an

advertisement of the routing peer, and an order sequence of gateways. This section contains a combined example to provide code to implement both sides of the endpoint routing conversation.

Chapter 10 A Complete Sample Application

This chapter guides the reader through the process of creating their own complete P2P solution implemented using JXTA, using all of the protocols. Examples from each of the preceding six chapters are used as the foundation of the application, and brought together to provide all of the elements of the final solution.

Chapter 11 The Future of JXTA

This chapter explores some of the future directions currently being investigated by the JXTA Projects, including implementations of JXTA for other languages, bindings to transports other than HTTP, and applications being built on top of JXTA. This section includes overviews of the other areas of interest within the JXTA project, including adding functionality to address security and wireless requirements, as well as support for other language and transport bindings.

Appendix A Glossary of Terms

This appendix provides a glossary of the technical terms and acronyms used in the book.

Appendix B Online Resources

This appendix provides the user with the addresses of web sites and mailing lists with related information on JXTA, and standards used by JXTA.