

Steal This Book!

Yes, you read that right. Steal this book. For free.

Nothing. Zero. Zilch. Nada. Zip.

Undoubtedly you're asking yourself, "Why would he give away a book he probably spent six grueling months writing? Is he crazy?"

The answer...is yes. Sort of. I know that every day you're faced with hundreds of computer titles, and a lot of them don't deliver the value or the information you need. So here's the deal: I'll give you this book (or this chapter, if you've only downloaded part of the book) for free provided you do me a couple favors:

1. **Send this book to your friends:** No, not your manager. Your "with it" computer friends who are looking for the next Big Thing. JXTA is it. Trust me. They want to know about it.
2. **Send a link to the book's web site:** Maybe the book is too big to send. After all, not everyone can have a fibre optic Internet connection installed in their bedroom. The site, at www.brendonwilson.com/projects/jxta, provides chapter-sized PDFs for easy downloading by the bandwidth-challenged.
3. **Visit the book's web site:** Being a professional developer, you probably have Carpal Tunnel Syndrome and shudder at the idea of typing in example source code. Save yourself the trouble. Go to www.brendonwilson.com/projects/jxta and download the source code. And while you're there, why not download some of the chapters you're missing?
4. **Buy the book:** You knew there had to be a catch. Sure, the book's PDFs are free, but I'm hoping that enough of you like the book so much that you have to buy a copy. Either that, or none of you can stand to read the book from a screen (or, worse yet, print it all out <shudder>) and resort to paper to save what's left of your eyesight. The book is available at your local bookstore or from Amazon.com (at a **handsome discount**, I might add).

I now return to your regularly scheduled program: enjoy the book!



12

The Future of JXTA

NOW THAT YOU'VE SEEN JXTA, you're probably wondering what your next step should be. Maybe you want to help Project JXTA refine the current Java reference implementation, or work on a reference implementation in your preferred development language. You might have a new project that you want to tackle with this new tool, or you might just want to see what other people are working on. This chapter provides information on where JXTA is heading and how you can participate.

Future Directions for Project JXTA

At several points in this book, you might have gotten the impression that JXTA is not currently a “complete” product. In truth, it might never be complete. Like many open-source projects, JXTA is constantly evolving as open-source developers augment the existing reference implementation and specification to address new problems in peer-to-peer computing.

Besides working on the Java reference implementation, developers within the JXTA Community are developing new services and applications built on top of the JXTA platform. The future success of JXTA depends on the capability of these applications and services to demonstrate the benefits and

viability of JXTA technology. In addition to working on applications of the JXTA technology, other developers are creating new reference implementations or the core JXTA platform in languages other than Java. These new reference implementations will enable developers in a variety of languages to recognize the benefits of JXTA in their preferred development language.

The following sections outline some of the most prominent services, applications, and new reference implementations being developed by the JXTA Community. This is by no means a comprehensive list, and it does not include applications that are being developed outside the JXTA Community hosted by Project JXTA. For the latest, look at the list of projects hosted by Project JXTA at www.jxta.org/servlets/DomainProjects.

Services

Services supply the building blocks that applications can use to provide real functionality to an end user. Services can be independent or can augment other services to provide new functionality. Some services might provide mechanisms of their own that replace the core services provided in the reference implementation, such as the Discovery service.

Nearly two dozen services projects currently are listed in the Services category of projects hosted by the Project JXTA site. These projects are in many different stages of development. Some projects are just beginning to do their initial design work, while others are working on coding implementations. This section outlines three of the most developed services within the JXTA Community. More information on services being developed by the JXTA Community is available from the Project JXTA Services web site, services.jxta.org.

JXTA Search

The JXTA Search project began at Infrasearch, a company that was later acquired by Sun Microsystems and folded into Project JXTA. Infrasearch's proof-of-concept technology used the Gnutella protocol to allow search clients to query information providers, such as web sites, to gain access to information not available via traditional search engines.

The JXTA Search project is building on that initial technology using JXTA to enable information consumers to search for information locked away in the "deep web." As outlined in Chapter 1, "Introduction," current search engines are limited to indexing static information provided on the web, resulting in

irrelevant or outdated responses. In addition, traditional search-engine technology does not capture or index information that is stored within corporate databases. As a result, search-engine query results are neither as comprehensive or as real-time as possible.

JXTA Search solves the problem by allowing information providers to integrate their corporate information stores with a distributed network of peers. The JXTA Search project defines a set of protocols built on top of JXTA that allows a client to query an information provider and obtain results from the information provider's store of information. The advantage of this approach is that the results of queries are more up-to-date and comprehensive than those obtained using search engines.

Note

JXTA Search doesn't necessarily replace current search-engine technology. Current search-engine technology is fairly well suited to the problem of indexing static pages. JXTA Search simply augments traditional search engines to provide access to information not captured by the indexing crawlers employed by search engines.

More information on the JXTA Search project can be found at the project's web site, search.jxta.org.

Content Management System (CMS)

The Content Management System (CMS) is designed to allow peers to share content with and retrieve shared content from other peers. The CMS service provides a foundation that other file- or document-sharing applications can use to handle the details of publishing and retrieving content from a set of distributed peers.

The CMS defines a Content Advertisement, which provides metadata describing a particular piece of content. The Content Advertisement itself includes an MD5 hash generated from the content data that uniquely identifies the content. This MD5 hash can be used by the peer to retrieve content from any peer hosting the content, without relying on parameters that could change, such as filename.

In addition to the advertisement definition, the CMS specifies a protocol for searching for and retrieving content. This protocol augments the basic JXTA pipe functionality to allow a peer to download content from a peer and ensure that the content is retrieved in a reliable fashion. Content shared by a peer is managed by the CMS in a persistent store containing the shared content and the content advertisements.

In the future, the CMS project hopes to augment the current search functionality to incorporate the JXTA Search service. More information on the project's progress can be found at the project's web site, cms.jxta.org.

JXTA-Remote Method Invocation (RMI)

The Remote Method Invocation API provided by the Java 2 Standard Edition allows a program to invoke methods on Java objects hosted on another JVM or even on a remote machine. By default, RMI uses TCP/IP as a network transport to invoke methods on remote objects and transports serialized object instances.

The JXTA-RMI service enables a Java developer to use JXTA pipes instead of TCP/IP as a transport mechanism. The advantage of this approach is that it allows existing RMI-based applications to build on JXTA without requiring major changes to the existing application. This implementation still provides only a point-to-point solution, despite JXTA's capability to handle many-to-many communication.

More information on RMI is available from Sun at java.sun.com/products/jdk/rmi/index.html, and current project information and source code for JXTA-RMI is available from the project's web site, jxta-rmi.jxta.org.

Applications

Applications build on existing JXTA services to provide an end user with some useful way of interacting with the JXTA P2P network. Usually the application incorporates some form of user interface, enables users to interact with JXTA services, and controls their behavior.

Currently more than a dozen application projects are listed in the Applications category of projects hosted by the Project JXTA site. One application being developed, the JXTA Shell, formed the basis of the examples in the first half of the book. This section outlines two of the most developed applications within the JXTA Community. More information on applications being developed by the JXTA Community is available from the Project JXTA Applications web site, apps.jxta.org.

myJXTA

Besides the JXTA Shell, myJXTA, shown in Figure 12.1, is probably one of the most fully featured JXTA applications. Originally called InstantP2P, myJXTA was one of the first applications developed to demonstrate the capabilities of the JXTA platform. This application is currently designed to run only on desktop computers, but it should eventually be capable of running on devices supporting the PersonalJava or J2ME platforms.

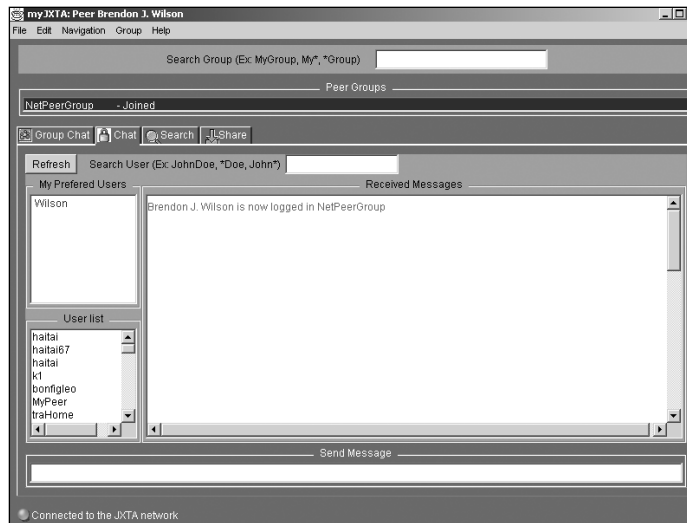


Figure 12.1 The myJXTA user interface.

The myJXTA application enables a user to participate in a group chat room, engage in one-to-one chat, share files, and create, join, and leave peer groups. The file-sharing functionality of myJXTA is built on top of the Content Management Service discussed in the previous section.

Because the myJXTA application is fairly stable, future plans include only fixing known bugs. The myJXTA source code is a good place to start when changes are made to the JXTA reference API because an updated version of the application is usually released at roughly the same time as a new stable release of the Java reference implementation. More information on the project and its source code can be found at the project's web site, instantp2p.jxta.org.

Gnougat

Gnougat, shown in Figure 12.2, is a file-sharing application similar to Gnutella, with the notable exception that the responsibilities of file sharing can be distributed across all peers in the network.

In traditional file-sharing applications, peers have no way to determine whether two search results point to the same content. Searches are performed based on the name of a shared file or metadata embedded in the file. Unfortunately, there is no guarantee that two files with matching metadata are identical. Gnougat attempts to solve this problem by focusing on content instead of metadata information.

In the Gnougat system, a file containing specific content is downloaded using a hash of the content itself. This allows the client peer to find peers hosting the same content and optimize the download of the file. In addition, the system allows client peers to split the download into multiple chunks, each of which can be downloaded in parallel from a different peer.

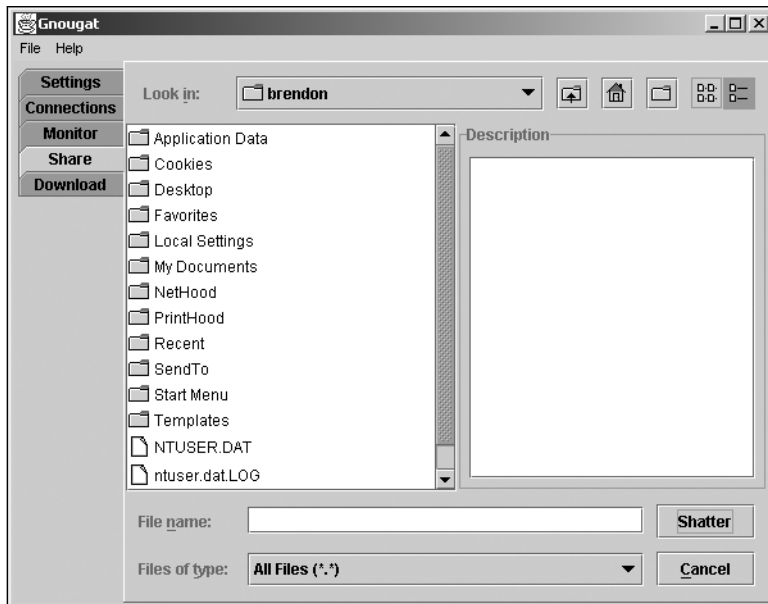


Figure 12.2 The Gnougat user interface.

The Gnougat application still uses metadata to search for shared files. Content is discovered initially using metadata, such as filename; then a hash descriptor for a particular piece of content is retrieved from the network. This hash descriptor is used by the client peer to perform the download by discovering peers hosting the exact same content.

A preliminary implementation of Gnougat has been made available by the developers. The installer, along with a discussion paper on the technology behind Gnougat, is available from the project's web site, gnougat.jxta.org.

Core Reference Implementations

A variety of non-Java reference implementations currently are in progress within the JXTA Community. Projects currently have been established for the C, Objective C, Perl, and Ruby languages.

Other implementation projects are focusing on producing JXTA implementations for small or constrained devices:

- The JXME project is producing a version of the JXTA platform suitable for handheld devices capable of running the Java 2 Micro Edition platform.
- The PocketPC project is also porting the JXTA reference implementation to C++ specifically for the PocketPC and CE platforms.
- The TINI project is working on a version of JXTA for the TINI platform, an embedded Java runtime designed to run on top of the DS80C390 microprocessor from Dallas Semiconductor.

At this point, most of these projects are in the very early stages of design and implementation. The Java reference implementation remains the most complete implementation of the JXTA specification at this time.

Participating in Project JXTA

The future of JXTA depends on developers contributing their skills to Project JXTA. As previously mentioned, JXTA is currently a work-in-progress, one that requires people to help with all aspects of the development. Whether you're interested in starting a new JXTA project or helping out with existing projects, Project JXTA will benefit from your involvement.

To start getting involved with Project JXTA, you should first do the following:

1. **Register.** If you are interested in contributing to an existing project or starting a project of your own, you should first become a member of the JXTA Community. Registration is free and can be done from the Project JXTA registration page at www.jxta.org/servlets/Join. Registering enables you to join projects as a contributor and propose projects of your own.
2. **Submit a contributor agreement.** To contribute code to Project JXTA, you must submit a contributor agreement. The agreement is available for download from www.jxta.org/jxta_contrib_agreement.PDF. This agreement ensures that the code you contribute is legally yours to contribute and is being contributed in accordance with the license employed by Project JXTA.
3. **Join the mailing lists.** The JXTA mailing lists are used to propose new projects, analyze problems to be tackled by peer-to-peer, and announce changes in the reference implementation. You should join at least the discuss@jxta.org mailing list to keep abreast of the latest JXTA developments.

The JXTA Mailing Lists

Project JXTA maintains a number of mailing lists to allow people to discuss aspects of JXTA and peer-to-peer technology. The four main mailing lists hosted by Project JXTA are listed here:

- **announce@jxta.org**—A list used to announce major releases to members of the general public who are interested in JXTA. This list is not meant to host discussions.
- **dev@jxta.org**—A list for developers working with JXTA to discuss technical issues related to JXTA. Currently, most of the discussion on this list focuses on the Java reference implementation of JXTA.
- **discuss@jxta.org**—A more general discussion list devoted to JXTA and the problems that need to be addressed by peer-to-peer systems. This list is also used to propose new projects to the JXTA Community and seek approval from the community.
- **user@jxta.org**—A list for new JXTA developers who are just starting to familiarize themselves with the reference implementation. Common themes include questions on using the JXTA Shell, using myJXTA, and performing basic operations with the reference implementation APIs.

Instructions for subscribing and unsubscribing from the mailing lists, as well as searchable archives of the lists, are available from www.jxta.org/project/www/maillist.html. As with any mailing list, you should monitor the mailing list for a short time before posting new messages, to ensure that you are posting content that is appropriate for the mailing list.

In addition to these four lists, each project hosted by Project JXTA hosts its own mailing lists. By default, each project hosts a `cvs` mailing list that announces when changes are committed to the project's source control and an `issues` mailing list that announces messages from the project's bug-tracking system. Although projects are free to create other discussion mailing lists, Project JXTA encourages projects to use the main `dev` and `discuss` mailing lists unless the project generates an unusual amount of mail. This ensures that the JXTA Community is apprised of developments in all projects currently under development.

Proposing a New JXTA Community Project

All this talk about the advantages of JXTA might have you thinking of developing a killer peer-to-peer application of your own. But before you run out and start development, consider proposing the project to the JXTA Community for any of the following reasons:

- **To obtain constructive criticism**—A lot of smart people are involved in the JXTA Community, and chances are good that some of them have considered the problem that your application or service will attempt to solve. Proposing the project to the community will allow discussion of the potential roadblocks that you will encounter and generate valuable feedback.
- **To eliminate duplication**—There's no point in starting a whole new project if there's currently a similar effort under way within the JXTA Community. Proposing the project to the community ensures that your application or service isn't a duplication or variation of an existing project. If there's a matching project in existence, you benefit by being able to join a team that has already done some of the work.
- **To form a team**—You don't want to do all this work yourself, do you? Proposing the project to the community gives you a chance to gather interested developers to help you implement your application.
- **To take advantage of tool hosting**—If your project is approved, you will be able to manage the project using the tools provided by jxta.org. These tools include hosted source control, bug tracking, and project mailing lists. These tools not only relieve you of the burden of hosting these tools yourself, but they also allow others to easily access and contribute to your project.

To propose a new project to the JXTA Community, you should first register with Project JXTA, submit a contributor's agreement, and subscribe to the discuss@jxta.org mailing list. After you have done this, send a message to the discuss@jxta.org mailing list with the project's proposal. The project proposal should contain the following:

- A descriptive subject line that includes the text "Proposed Project:" and the name of the proposed project.
- A complete description of the project and the particular problem that the project will attempt to solve.
- A description of the project category. The categories correspond to those on the projects page: Apps, Core, Demos, Other, or Services.

You must ensure that your project will conform to the terms of the license employed by Project JXTA. License information is available from www.jxta.org/license.html.

After you have posted a proposal message to the mailing list, you should begin receiving feedback indicating whether the JXTA Community thinks that your project would be worth undertaking. While this discussion is occurring, you should also register with Project JXTA, create the project's home page, and formally propose the project to the jxta.org Community Manager using the form at www.jxta.org/servlets/ProjectAddStep1. If your proposal is complete and the discussion on the mailing list indicates that there is reason to pursue the proposed project, the Community Manager should approve your project after a few days.

Working with the Java Reference Implementation Source Code

The Java reference implementation is the most active project currently under way in the JXTA Community, and many other projects build on its capabilities. Because most of the Java-based projects have a common build system, this section shows you how to obtain the latest version of the reference implementation and build it.

Obtaining Java Reference Implementation Source Code

As previously mentioned, Project JXTA provides hosted projects with a source-control system to use to manage the project's source code. The source-control system used by Project JXTA is the Concurrent Versions System (CVS), which allows multiple developers to work on the same code simultaneously and merge their work.

To access the CVS source repository, you'll need to install a CVS client. For the demonstration, I'll be using WinCVS, a Windows-based CVS client available from www.wincvs.com that provides a simple user interface. Non-Windows and command-line CVS clients are available from www.cvshome.org/downloads.html. Manuals and information on CVS are also available from www.cvshome.org.

To obtain the Java reference implementation, follow these steps:

1. Start WinCVS.
2. Select the Checkout Module item under the Create menu. The Checkout Settings dialog box displays (see Figure 12.3).

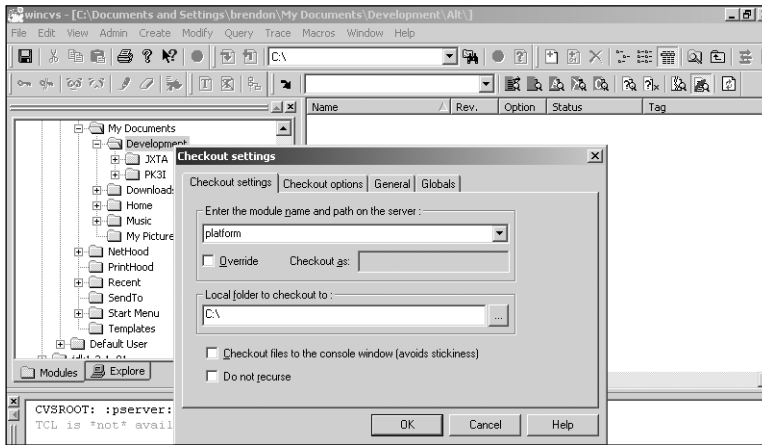


Figure 12.3 The Checkout Settings dialog box.

3. Enter **platform** in the Checkout Settings dialog box. This is the CVS module name of the Java reference implementation in the source code repository.
4. Select the local folder where the retrieved source code will be placed.
5. Go to the General tab.
6. Enter **:pserver:guest@cvs.jxta.org:/cvs** into the text field for the CVSROOT. This specifies the user to use to retrieve the source code. For this example, just use the guest user.
7. Ensure that Authentication is set to Password File on the CVS Server.
8. Click OK.

The CVS client should now contact the server, retrieve the source code for the Java reference implementation, and place it in the specified local directory. When the CVS client has completed the download from source control, you can build the downloaded source code.

Building the Java Reference Implementation Source Code

The Java reference implementation, as well as many of the Java-based JXTA Community projects, use the Ant build tool created by the Apache Jakarta project (jakarta.apache.org). To build the reference implementation, you first should download and install the latest version of the Ant build tool from the Ant home page at jakarta.apache.org/ant/index.html.

Ant provides a build tool similar to *make*, but it is based on Java and is therefore cross-platform. To use Ant, you need only to install Ant and add the path to Ant's bin to your system path. On some systems, you might also need to set a `JAVA_HOME` variable to point to the location of the JDK.

To build the reference implementation, follow these steps:

1. Open a command prompt.
2. Change to the directory containing the reference implementation code that you retrieved from source control.
3. Change to the `binding\java` subdirectory.
4. Set the `JAVA_HOME` variable. On Windows, this is achieved by typing **SET JAVA_HOME=** followed by the fully qualified path to your JDK (for example, `C:\jdk1.3.1_01\`).
5. Include the Ant bin directory in your system path. On Windows, this is achieved by typing **SET PATH=%PATH%;** followed by the fully qualified path to the Ant bin directory.
6. Build the source code by typing **ant**.

The Ant build tool will use the `build.xml` file to compile all the Java reference implementation source code and create the `jxta.jar` file. The compiled classes will be placed in the `classes` subdirectory, and `jxta.jar` will be placed in the `lib` subdirectory.

Summary

In this chapter, you learned about a few of the services, applications, and implementations of the JXTA technology, and you learned how to get involved with Project JXTA to help shape the future of the technology. The rest, to paraphrase my university professors, “is left as an exercise for the reader.”

This book has been about teaching people how to use JXTA so that they can go out and define the direction of peer-to-peer technology. I hope that this book has helped give you the tools that you need to go out and produce your own solutions. If you think that something should be included or expanded in future versions of the book (assuming that there is a second edition), feel free to drop me an email via my web site, www.brendonwilson.com.