

# Project JXTA: Getting Started

Getting started with JXTA is simple. If you're reading this, you probably have already been to the Project JXTA website at <http://www.jxta.org>. If not, check it out first, it's the center of activity for Project JXTA, with documentation, FAQs, specifications, whitepapers, downloads, and discussions about peer-to-peer (P2P) computing and the expanded Web.

## Get the JXTA Platform Binaries

Perhaps the quickest way to appreciate what's possible with peer-to-peer computing is to download Version 1.0 of the JXTA Shell. The JXTA Shell supports these systems running the Java™ Runtime Environment: Microsoft Windows, Linux, Apple Macintosh, or the Solaris™ Operating Environments. Specific requirements include:

- Microsoft Windows 98, Windows ME, or Windows 2000 with Internet Explorer V5.5 or later; or the Java Runtime Environment Version 1.1.4 or later, 5 MB of hard disk space, and a network connection to the Internet.
- All other systems (Linux, Macintosh, and Solaris Operating Environment) require the Java Runtime Environment Version 1.1.4 or later. Additional requirements are 5 MB of hard disk space and a network connection to the Internet.

To download the software, click on the *Download JXTA Shell Now* link on the Project JXTA home page. Here you can select the version of the JXTA Shell's installation program appropriate for your system. Get the appropriate version and run the downloaded file (e.g. *JXTAInst.exe* on Windows) and follow the on-screen instructions to complete the installation.

Developers wishing to experiment with and build the JXTA platform or the JXTA Shell can also download the source. See the Project JXTA home page for additional information.

## Running the JXTA Shell

For Microsoft Windows users, the JXTA Shell can be run from the Start Menu at: Start->Program Files->JXTA->JXTA Shell.

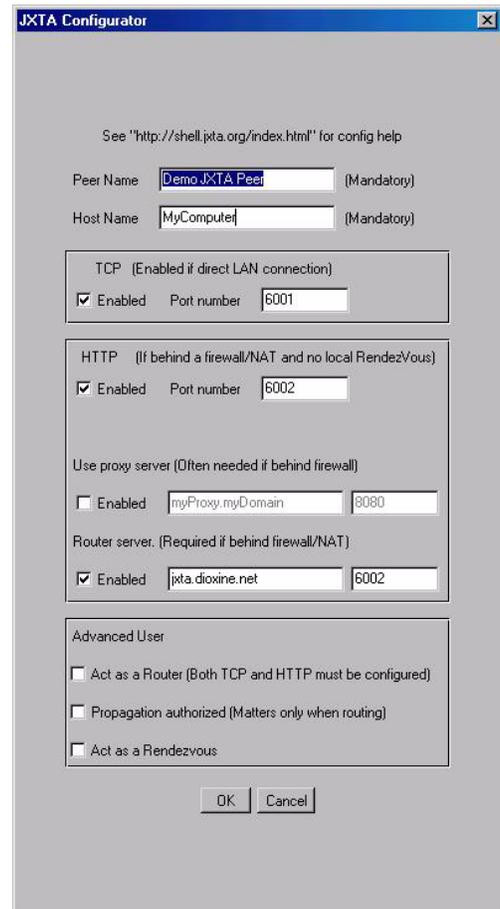
The first time the JXTA Shell is run, a configuration screen will appear (see figure at right.) This screen allows you to set up the software to operate with your network. Most users will only have to enter a nickname in the top field and leave the rest of the settings at their default values. Additional settings may be adjusted for users behind firewalls or those wishing to configure for advanced operation. Additional information about the JXTA Shell can be found in the README file included with the installation *Jxta\_Readme.html* or on the Project JXTA web site at <http://shell.jxta.org/release.html>. General information about the JXTA Shell can be found at <http://shell.jxta.org/index.html>.

If you think you may be behind a firewall but are not sure of the proper settings, check the advanced property settings for your web browser and copy its proxy settings into the JXTA Configurator's *Proxy Server* field. Contact your system administrator if you need additional information about your network environment.

## Experiment with the JXTA Shell

By experimenting with the JXTA Shell, you'll quickly grasp the power and potential of P2P. While the JXTA Shell is simply a P2P application built on top of the JXTA platform, it's much more than that. The JXTA Shell is a command interpreter that also demonstrates key concepts of the JXTA platform including peers, peer groups, pipes, security, and monitoring.

The JXTA shell provides several commands to discover and manage membership in peer groups.



JXTA Configurator

See "<http://shell.jxta.org/index.html>" for config help

Peer Name:  (Mandatory)

Host Name:  (Mandatory)

TCP (Enabled if direct LAN connection)

Enabled Port number:

HTTP (If behind a firewall/NAT and no local Rendezvous)

Enabled Port number:

Use proxy server (Often needed if behind firewall)

Enabled

Router server: (Required if behind firewall/NAT)

Enabled

Advanced User

Act as a Router (Both TCP and HTTP must be configured)

Propagation authorized (Matters only when routing)

Act as a Rendezvous

OK Cancel

The core commands include:

- *man* to obtain help via an on-line “manual” and to list all the available commands
- *exit* to exit the Shell
- *whoami* to display peer and peer group information
- *peers* to discover and list peers
- *groups* to discover and list peer groups
- *join* to join a peer group
- *leave* to leave a peer group
- *talk* allows peers to pass text messages back and forth similar to an instant messaging application

There are other commands that offer more fine-grained control and functionality. You can learn about them at <http://www.jxta.org/ShellQuickGuide.html>.

### Entering Commands to the JXTA Shell

Once you have completed configuration of the JXTA Shell, it will respond with a welcome message and prompt (see figure at right.)

```
JXTA>
```

### Getting Help on the Shell Commands

The *man* command is the primary help system for the JXTA Shell. The usage of *man* is:

```
JXTA> man <commandName>
```

For instance, typing

```
JXTA> man Shell
```

displays help about the Shell

### Discovering other Peers

The *peers* command is used to discover other peers. Without an option, the *peers* command shows only the list of peers that have been previously found, no remote discovery is performed.

The first time the *peers* command is run only the local peers or pre-configured rendezvous peers are shown:

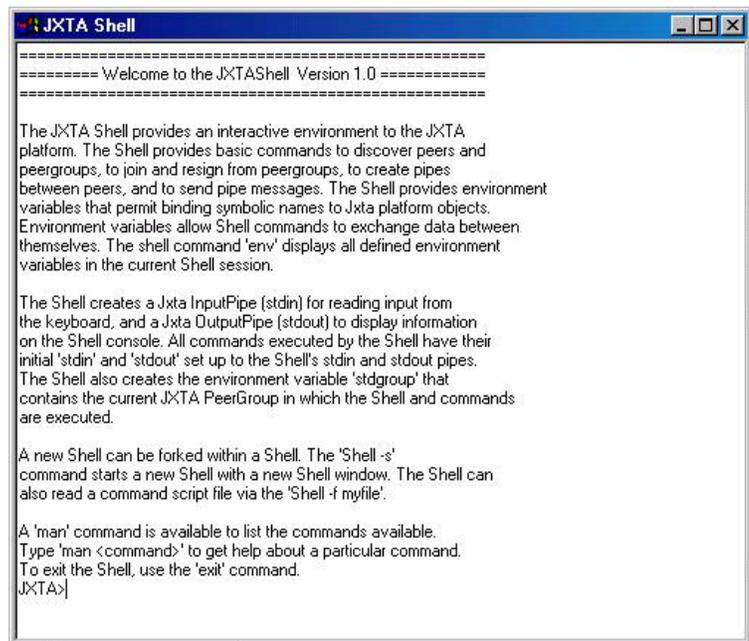
```
JXTA> peers
peer0: name = Demo JXTA Peer
```

This lists only the current peer named *Demo JXTA Peer*. To find other peers, a remote discovery must be performed via the *-r* option. The command is run asynchronously and returns with a prompt without waiting for a peer to be discovered.

```
JXTA> peers -r
peer discovery message sent
JXTA>
```

The *peers* command can then be issued again any number of times to list the new peers that have been discovered. Depending on the network latency, peer discovery can be instantaneous or take few minutes. Peer discovery on slow transports such HTTP or when a remote connection involves lot of hops in case of firewalls or NAT configurations will take more time.

```
JXTA> peers
peer0: name = Demo JXTA Peer
peer1: name = Dioxine.net
peer2: name = chrisatwork
peer3: name = ecksteinpeer
peer4: name = emily5
peer6: name = john@home
JXTA>
```





```
<Service>jxta.service.resolver</Service>
<Service>jxta.service.peerinfo</Service>
<Service>jxta.service.membership</Service>
JXTA>
```

The *join* command will now show the current information about the peer group joined by this peer.

```
JXTA> join
Joined Group   : stevesgroup (current)
JXTA>
```

A peer group can be created using the *mkadv* and *mkgrp* commands.

```
JXTA> mygroupadv = mkadv -g MyNewGroup
JXTA> mygrp = mkgrp -d mygroupadv MyNewGroup
```

To verify your new group has been created, you can use the *groups* command:

```
JXTA> groups
group0: name = NetPeerGroup
group1: name = MyNewGroup
JXTA>
```

Assuming your peer is connected to the Internet, information about your new group will be propagated to other JXTA nodes on the Internet. It will take an indeterminate time for information about your new group to propagate among the JXTA peer nodes.

Now you can join your group:

```
JXTA> join -d group1
JXTA>
```

You can invite others to join your peer group by sending a message with your new group's name. Others can initiate a search to find the new group using a qualified group search command.

```
JXTA> groups -r -aName -vMyNewGroup
JXTA>
```

Note that this query can take some time to complete assuming there is a connection of JXTA peers between the searching peer and any peer aware of the new group. It may require several attempts before the groups are found.

### Communicating with other Peer Nodes

The *talk* command implements a command that is like a simple instant messaging application, allowing two peers to exchange text messages. Messages are displayed on the Shell console. In order to use *talk*, the user must register first. This is done by registering using the *talk* command and the *-register* option. This creates a talk advertisement for the user. This need be done only once, the first time the user registers with *talk*. The system remembers the registration from then on.

```
JXTA> talk -register moi
User moi is now registered
JXTA>
```

Login using the *talk* command and the *-login* option. This command logs the user and starts a listener daemon and creates an input pipe enabling others to send messages. This command has to be issued everytime the peer is restarted.

```
JXTA> talk -login moi
JXTA>
```

The user can now talk to another user via *talk* command. The *-u* option allows the peer initiating the talk session to specify their peer identity. The *talk* command below will search for a listening peer for user *MyFriend*. If one is found, *talk* prompts the user to enter a message to send. In order to talk to another user, one must have obtained the other users login name. This needs to be done independently by email or phone, or by using the *search* command.

```
JXTA> talk -u moi MyFriend
talk is connected to user MyFriend
Type your message. To exit, type '.' at beginning of line
Hi Mike - How about meeting me for lunch at 12:30. My office?
...
JXTA>
```

If you would prefer to search for other users automatically, use the *search* command. In the example below, advertisements for 6 users were found. Looking at the contents of one of the shell variables, *adv1*, reveals a username of *toolbox*, which could be used in the *talk* command.

```
JXTA> search -r -aname -vJxtaTalk*
JXTA Advertisement search message sent
JXTA> search
JXTA Advertisement adv0
JXTA Advertisement adv1
JXTA Advertisement adv2
JXTA Advertisement adv3
JXTA Advertisement adv4
JXTA Advertisement adv5
JXTA> cat adv1
<?xml version="1.0"?>
<!DOCTYPE jxta:PipeAdvertisement>
<jxta:PipeAdvertisement>
  <id>

jxta://
59616261646162614A757874614D50475749FC3372C14DCA9E7EAE427863456C000000000000
00000000000000000000000000000000000000000000000000000000000000000000401
  </id>
  <name>
    JxtaTalkUserName:toolbox
  </name>
</jxta:PipeAdvertisement>
JXTA>
```

Use the *man* command to get more information about *search*.)

To stop receiving any more *talk* messages, the user can stop the *talk* daemon by entering the command:

```
JXTA> talk -logout <username>
JXTA>
```

### Monitoring JXTA Peer Nodes

The JXTA platform has facilities to monitor local and remote peer nodes. The *peerinfo* command instructs a peer to store its current status information. To store and view your peer's current local information:

```
JXTA> peerinfo -l
stored peerinfo0
JXTA> cat peerinfo0
<?xml version="1.0"?>
<!DOCTYPE jxta:PeerInfoAdvertisement>
<jxta:PeerInfoAdvertisement>
  <sourcePid>

jxta://
59616261646162614A78746150325033C5D7371A5F374A6899429C21D077EDE1000000000000
00000000000000000000000000000000000000000000000000000000000000000000301
  </sourcePid>
  <targetPid>

jxta://
59616261646162614A78746150325033C5D7371A5F374A6899429C21D077EDE1000000000000
00000000000000000000000000000000000000000000000000000000000000000000301
  </targetPid>
  <uptime>
    411
  </uptime>
```

```
.....  
</jxta:PeerInfoAdvertisement>  
JXTA>
```

In the truncated example above, you can see peer uptime and peer IDs. Updated information may be viewed by repeating the command sequence.

## Develop Some Applications

After playing with the JXTA Shell, we're willing to bet that you'll agree that this P2P stuff is pretty cool. If you're ready to know more, a good place to start is the whitepaper entitled *Project JXTA: A Technology Overview*. To know the details about JXTA protocols and advertisements, the *Project JXTA: Technical Specification* is a must read. And of course, there's no substitute for looking at code. You'll find it all at the Project JXTA home page.

The source code and specifications for Project JXTA are posted under the Sun Project JXTA Software License, which is based on the Apache Software License Version 1.1. Source code is free for you to use according to the license.

As with any open source project, our goal is to get the developer community involved by contributing to Project JXTA. There are many ways of contributing to an open source software project:

- *File a bug report.* Detailed bug reports are valuable. Next time you come across something you think is a bug, take some time to reproduce it. If you can, file a bug report by cutting and pasting the relevant information from your log file. You can file bug reports using *Issuezilla*, the Project JXTA bug-tracking system.
- *Join a mailing list.* Post general feedback, request features, and ask for help. Also, if you see a question on the list that you can answer, get to it! See the mailing lists page on the Project JXTA website for details on how to subscribe.
- *Write an application and share it.* Peer-to-peer computing is the next wave, but people will need to get their hands on an application that does something useful to fully appreciate its potential. We're willing to bet that after you play with the JXTA Shell and look at the documentation, you'll think of a million cool applications.

*Join us.* We welcome your feedback, comments, and code.

SUN MICROSYSTEMS, INC., 901 SAN ANTONIO ROAD, PALO ALTO, CA 94303-4900 USA  
PHONE: 650-960-1300 FAX: 650-969-9131 INTERNET: [www.sun.com](http://www.sun.com)



©2000 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, Sun Enterprise, Solaris, and SunSITE are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc., in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, LTD. All trademarks and registered trademarks of other products and services mentioned in this report are the property of their respective owners.

Draft 1.0 April 25, 2001