



FastPak for Java

Frequently Asked Questions

Questions

1. [What's the difference between FastPak and WebStart?](#)
2. [What's the Controller Thread?](#)
3. [What's the difference between console, GUI, and daemon modes?](#)
4. [How many instances of FastPak may a single user run on their system?](#)
5. [How does FastPak conserve memory?](#)
6. [How does FastPak performance compare to applications launched with WebStart?](#)
7. [How does FastPak performance compare to running an application with its own JVM?](#)
8. [Can I launch Tomcat or other Java based servers under FastPak?](#)
9. [Can I run JNI applications under FastPak?](#)
10. [How difficult is it to set up an application to run under FastPak?](#)
11. [Does FastPak use XML in its configuration files?](#)
12. [What are the system requirements for FastPak?](#)
13. [I'm running a FastPak version built for Java 1.6. Can I run applications built with earlier Java releases?](#)
14. [Do all applications have to be in jar format?](#)
15. [If I use FastPak, does that mean all Java applications I have must be executed with FastPak?](#)
16. [Can I use the FastPak binaries built for one operating](#)

[system release, like Windows, on a different operating system like Linux?](#)

17. [Can I build and release custom GUIs for FastPak and sell them or release them as open source?](#)
18. [Do applications need to be specifically built for FastPak?](#)
19. [Since all applications running under FastPak are running under a single JVM, won't they interfere with one another?](#)
20. [Can FastPak automatically launch applications at start up?](#)
21. [Why can't the Controller Thread be used as a general network interface?](#)
22. [How can FastPak make applications launch faster than they do when I launch them with a shell script?](#)
23. [I thought with todays contemporary operating systems and shared resources, JVM consumption of resources is way down. If this is true, then how can FastPak save system threads and memory as you claim?](#)
24. [Since FastPak can launch applications in a jar file hosted on a remote server, how difficult is it to configure this type of set up? Is it secure?](#)
25. [If I don't use the Controller Thread, should it be shut down?](#)
26. [Is FastPak open source?](#)
27. [Why must I use the installer? What does it do?](#)
28. [Since FastPak and any applications running under it use a single JVM, if FastPak is shut down, won't all the applications die at the same time?](#)
29. [Can FastPak launch a native application?](#)
30. [Can FastPak launch another JVM external to FastPak?](#)
31. [Can on running FastPak instance launch other FastPak instances?](#)
32. [We have written two Java based servers applications. Each one establishes its own connection with its own port. Won't this cause some type of internal conflict with FastPak?](#)
33. [About how much time do you think it would take a developer or administrator with a few years of experience to learn to use FastPak?](#)
34. [Since FastPak is released base on Java versions, which one is recommended? Which ones come with the](#)

[distribution kit?](#)

35. [When I get on your site, some of our security software throws up a warning and indicates your web site is attempting to record something. What's going on here?](#)
36. [Why isn't FastPak available outside the U.S.?](#)
37. [We want to order more license copies of FastPak than your web site advertises. Can we do this and receive further discounts?](#)
38. [Will FastPak be ported to other operating systems besides Windows, Linux, and OS X?](#)
39. [We have several server applications running on a server. Because of your licensing policies, our understanding is that if we wish to let each one of these run under a unique FastPak instance and take advantage of the Controller Threads ability to shut dow](#)
40. [We don't like the FastPak GUI supplied with the product. Our end users are not technical and we keep getting calls to the support group about threads dying, which is scaring some of the users. We've gone through the developer documentation and realize the](#)
41. [Why is the password option disabled when configuring FastPak?](#)
42. [We noticed that some applications can end or terminate under FastPak, but messages indicating they've terminated may not appear sometimes for hours. Why is this?](#)
43. [We used the FastPak GUI to delete some jarred up applications because they were no longer needed. The profiles are gone from the profiles directory, and they no longer show up as viable applications to the end users, but the applications in jar format are](#)
44. [Why isn't FastPak available for Java versions prior to 1.4?](#)
45. [Your development manuals state that programs must avoid the "System.exit\(\)" call. I don't understand this. I have a ton of development books dating back to Java 1.1 from renowned authors on the subject that say using System.exit\(\) is a normal way to end a](#)
46. [I find the necessity of having to modify the change from GUI to daemon and back to GUI mode really pretty tedious because I need to open up config file and edit them manually in between. Are you going to change this?](#)
47. [The sales section of your site offers a much lower cost](#)

[“unsupported” license. Why would anyone buy this?](#)

Answers

1. What's the difference between FastPak and WebStart?

Night and day...how's that for an answer? WebStart is a general launching tool that's typically web based whereas FastPak runs as a program with either a GUI, command line console, or daemon (no interface). WebStart launches a new JVM for each application launched whereas FastPak runs all applications under a single JVM, thus consuming far fewer resources. WebStart offers no means to remotely control an application running on a remote machine, whereas FastPak can offer a degree of control via its networked Controller Thread. WebStart cannot be used to launch native applications, whereas, as of FastPak version 1.2.10.55, Update 3 FastPak can do this with an embedded tool.

We do not see FastPak as a competitor or even a replacement for WebStart. FastPak is a different beast altogether. About the only similarity the two share in common is that they can both launch applications, and that's about it.

2. What's the Controller Thread?

The Controller Thread is a thread inside the FastPak kernel that enables remote communication and control of a running FastPak instance. This means a remotely accessed FastPak instance can start new Java and native applications on the remote machines, terminate the FastPak instance, and report data and statistics on FastPak and all the Java based applications it has running underneath it. The Controller thread uses a simple sockets based API and the full kits come with example programs that can be used to control FastPak via command line (terminal) windows and GUIs.

3. What's the difference between console, GUI, and daemon modes?

GUI mode is the default mode and it presents users with a GUI that's easy to use and learn. The console mode is really intended to be used by people accessing a machine remotely via a terminal window (such as an X-Terminal session on Linux), and Daemon

mode provides no interface at all.

4. How many instances of FastPak may a single user run on their system?

As many as they want. The only restriction is that if the Controller Thread is active, each FastPak instance must have it's own unique port ID. If the Controller Thread is not (and won't be) active then even this can be bypassed. Some of our customers have used several different instances of FastPak to "bundle" different sets of applications and then autolaunch them in daemon mode when the instance of FastPak is started.

5. How does FastPak conserve memory?

FastPak by itself doesn't. FastPak will start saving memory once more than one application is run under it. When this occurs, some of the redundant resource consumption that occurs when multiple JVMs are started is eliminated. This saves on memory, threads, processes, etc.

6. How does FastPak performance compare to applications launched with WebStart?

This can vary from machine to machine, OS to OS, graphics card to graphics card, and Java version to Java version, but generally the performance is comparable. Since FastPak only uses a single JVM it can reduce the amount of "interference" JVMs have often introduced when multiple Java applications are launched.

7. How does FastPak performance compare to running a single application with its own JVM?

If only one application is launched, FastPak will introduce an overhead for the GUI (if used), plus some other resources in terms of threads and memory, but it's generally not, in our opinion, significant, and performance should generally be as good or very close to it. If you have one and only one Java application that's going to be run, the only reason to use FastPak in such a case would be to take advantage of the remote control features FastPak can offer via the Controller Thread. Some of our clients are doing this with Java based servers because FastPak can immediately take down the application on demand in the event of things like denial of service attacks (keep in mind, the Controller Thread uses it's own dedicated communications channel) or on experimental (as in under development) programs that may cause system level problems if not terminated quickly.

8. Can I launch Tomcat or other Java based servers under FastPak?

We haven't tried this yet, but our suspicion that yes, it could be done. It would require a considerable amount of extraction of the scripts used to start up Tomcat to get it to run. FastPak is very frequently used to launch stand alone Java servers and applications because the Controller Thread can allow them to be brought down in an emergency very rapidly.

9. Can I run JNI applications under FastPak?

Yes, with restrictions. Generally only one should be run at a time for a given instance of FastPak because attempting to launch a second one can cause FastPak to throw exceptions as it tries to load a native library that's already been loaded.

10. How difficult is it to set up an application to run under FastPak?

Most people familiar with application set up can learn to configure an application to run in less than an hour after installing FastPak. FastPak is very easy to use.

11. Does FastPak use XML in its configuration files?

No. FastPak configuration files are commented with instructions when configured using FastPak tools, and we saw no need to put the additional overhead of XML parsing into the product. FastPak is performance oriented.

12. What are the system requirements for FastPak?

The same as those for running any Java applications on a system. However, FastPak requires that the JVM be associated with a 1.4 release or later, and the FastPak releases are Java version specific, meaning that if you use a version of FastPak compiled for Java 1.4, you can't use it for applications built using Java 1.6. We strongly recommend using the latest release of both FastPak and Java for your systems and be sure the two are compatible.

13. I'm running a FastPak version built for Java 1.6. Can I run

applications built with earlier Java releases?

Generally, yes. The only exceptions to this rule will be versions that make use of outdated features (like killing threads) that have been superseded since Java 1.1. or applications that, for some reason, require a very specific version of Java. FastPak can, however, be configured to run an old JVM as a native application, but such an application is seen by FastPak as a native application and once executed it's out of FastPak's control and monitoring capabilities.

14. Do all applications have to be in jar format?

No. FastPak can be easily configured to run an unjarred application as well. In fact, the demo application supplied with all kits that uses the profile name "ActiveLogo" is such an application. If such an application is added outside of FastPak's normal class path settings, the paths will need to be added to FastPak and FastPak will need to be restarted. This is documented in both the developer and users manuals.

15. If I use FastPak, does that mean all Java applications I have must be executed with FastPak?

No. FastPak can reside side by side with applications launched with discrete JVMs or under WebStart. FastPak does not mandate everything be run under FastPak.

16. Can I use the FastPak binaries built for one operating system release, like Windows, on a different operating system like Linux?

Generally, no. FastPak is built for specific operating systems.

17. Can I build and release custom GUIs for FastPak and sell them or release them as open source?

Yes. In fact we encourage this.

18. Do applications need to be specifically built for FastPak?

It depends on what the application does, and more specifically, whether or not it terminates the JVM on exit. This is detailed in the developers manual. Because FastPak uses a single JVM to launch multiple applications, if a program issues a call, such as

System.exit(0) that terminates the JVM, then FastPak will terminate along with all other programs running underneath it. The use of System.exit() calls (and equivalent) was actually a long standing bug in the JVMs that was fixed in Java 1.4 and later. This is why FastPak requires the use of Java 1.4 or greater. This is detailed in the developers manual as well as in the Java documentation. In many cases, even this isn't a problem, and some people actually use it to their benefit if they're aware of the fact that FastPak will shut down when such system calls are issued.

19. Since all applications running under FastPak are running under a single JVM, won't they interfere with one another?

No. The only problems a user may encounter might be resource consumption which would likely still occur had the applications be run under dedicated JVMs.

20. Can FastPak automatically launch applications at start up?

Yes. In addition FastPak allows users to set priorities of applications with respect to one another. Many users use this in daemon mode and launch "bundled" applications all at once under several different FastPak configurations.

21. Why can't the Controller Thread be used as a general network interface?

The Controller Thread was designed to operate completely outside of a given application's environment, and it was designed to use a high speed, very simple protocol so it could control the FastPak kernel as quickly as possible. Allowing the Controller Thread's socket to be used as a general communications link could conceivably load the channel and slow messages coming from a remote access point. If a remote user is relying on FastPak to bring down an application in an emergency situation, this could end up blocking or delaying the message from being received in a timely manner. In addition, one of the design requirements of FastPak was that any application that FastPak could launch would also launch and execute completely outside of the FastPak environment (i.e. Running an application by double clicking on it or via WebStart). Allowing users to use a FastPak "specific" API would violate this requirement since such applications.

22. How can FastPak make applications launch faster than

they do when I launch them with a shell script?

Each FastPak instance loads one and only one JVM. Once loaded, when an application is executed, it's not a matter of having the operating system load and run yet another instance of the JVM, it's simply a matter of loading all the byte codes comprising the application.

23. I thought with today's contemporary operating systems and shared resources, JVM consumption of resources is way down. If this is true, then how can FastPak save system threads and memory as you claim?

That's true and false. Yes, libraries are shared and things can be cached, but it needs to be remembered that each instance of a JVM started is in effect a "sub-operating system" in and of itself, with an enormous amount of overhead. Each time a user launches a new application by typing "java <some application name>", using WebStart, or double clicking on a jar file, the operating system must re-allocate a considerable amount of resources over and over again. This is why on many machines as more and more applications are launched the load times increase. It's a simple matter of resource consumption, or as some might say, resource abuse. Each instance of FastPak typically uses a single JVM instead of dedicating a new JVM for each application launch, hence resources are minimized and load times are reduced.

24. Since FastPak can launch applications in a jar file hosted on a remote server, how difficult is it to configure this type of set up? Is it secure?

Very easy. All one needs to do is put the jarred application on a web server in a directory and then configure the application under FastPak. As of FastPak version 1. 2.10.55, Update 3, FastPak also includes a module that can check jar files signed with a private key against a public key.

25. If I don't use the Controller Thread, should it be shut down?

Yes. There's no need to leave a socket open if it's not in use.

26. Is FastPak open source?

No, and it's unlikely that the FastPak kernel will ever be released as open source code due to its complexity. We do, however,

encourage independent developers to produce both commercial and open source interfaces, particularly GUI and remote interfaces for the product.

27. Why must I use the installer? What does it do?

Yes, you must use the installer. The installer configures all of FastPak's configuration files based on its installed directory as well as the hosting operating system.

28. Since FastPak and any applications running under it use a single JVM, if FastPak is shut down, won't all the applications die at the same time?

Yes and no. FastPak offers two options to terminate it. One is immediate shutdown where FastPak and all applications it's launched will die immediately, and the other is an "exit" mode where all interfaces to FastPak will be shutdown, the kernels threading activities will be minimized, and application execution will be optimized and will terminate when the last running application finally ends.

29. Can FastPak launch a native application?

As of FastPak version 1.2.10.55, Update 1, there is a native application launcher included with FastPak. It should be noted that any native applications launched outside of a FastPak environment are not under FastPak's control and will run independently of FastPak once successfully started.

30. Can FastPak launch another JVM external to FastPak?

Yes, but it will be launched as a native application and reside outside of FastPaks control.

31. Can one running FastPak instance launch other FastPak instances?

No.

32. We have written two Java based servers applications. Each one establishes its own connection with its own port. Won't this cause some type of internal conflict with FastPak?

No. The only thing you need to worry about is whether the selected ports conflict with any of FastPak's ports or any other ports that may already be in use.

33. About how much time do you think it would take a developer or administrator with a few years of experience to learn to use FastPak?

To use FastPak it should take an hour or two. To understand some of the programming caveats associated with FastPak we would estimate, depending on experience, to be between 8 and 24 working hours. This is our opinion, not a documented or analyzed fact!

34. Since FastPak is released base on Java versions, which one is recommended? Which ones come with the distribution kit?

Commercial releases currently always include releases for all major releases of Java for that operating system. As of July, 2007, this means all purchased kits will include releases for Java 1.4, 1.5, and 1.6 for Windows and Linux, and 1.4 and 1.5 for OS X. We recommend you use the latest version possible, which is 1.6 for Windows and Linux, and 1.5 for OS X.

35. When I get on your site, some of our security software throws up a warning and indicates your web site is attempting to record something. What's going on here?

We had a real time statistics monitoring package installed some time ago. We are removing it from our web pages because too many people were afraid we were installing something on there systems. We aren't.

36. Why isn't FastPak available outside the U.S.?

There are two reasons for this. First, our billing system currently isn't set up for international accounts. Second, we're exploring the use of cryptological interfaces that may require U.S. Department of Commerce review and approval before being implemented, if we choose to implement them at all.

37. We want to order more license copies of FastPak than your web site advertises. Can we do this and receive further discounts?

Yes. Send a request to sales@scsc-online.com .

38. Will FastPak be ported to other operating systems besides Windows, Linux, and OS X?

This will be done on an as-needed basis. Our developers can, however, develop specific ports for unsupported operating systems, but consulting and development fees will apply.

39. We have several server applications running on a server. Because of your licensing policies, our understanding is that if we wish to let each one of these run under a unique FastPak instance and take advantage of the Controller Threads ability to shut down any number of these on demand (like during a denial of service attack) only the person the product is licensed to can legally access it. This seems sort of stupid to us. Isn't there a work around?

As of FastPak release 1.2.10.55 Update 3, we will be offering a "server" license to address this issue. It will cost a bit more but it will still be inexpensive.

40. We don't like the FastPak GUI supplied with the product. Our end users are not technical and we keep getting calls to the support group about threads dying, which is scaring some of the users. We've gone through the developer documentation and realize the message is just reporting garbage collection of dead objects. We would like to replace the GUI with one of our own using the techniques described in the manuals. If we do this, will we need to redistribute the API library? Is this allowable?

No, the API library is part of all FastPak kits. All that needs to be done is interface your new GUI with the API and make sure the Java versions are all correct. You should not distribute the API library with your kit because it may be out of date. All API libraries are upwardly compatible.

41. Why is the password option disabled when configuring FastPak?

This is a leftover from the first release of FastPak, but we're considering implementing it again, but with appropriate cryptological processes.

42. We noticed that some applications can end or terminate under FastPak, but messages indicating they've terminated may not appear sometimes for hours. Why is this?

This is due to the garbage collection mechanism of the JVM. Garbage collection under a JVM is unpredictable and it's generally like this on all operating systems.

43. We used the FastPak GUI to delete some jarred up applications because they were no longer needed. The profiles are gone from the profiles directory, and they no longer show up as viable applications to the end users, but the applications in jar format are still there. Why?

This is by design. People frequently delete things by accident, or do so and then realize later on that they need that application. We decided to simply delete the profile and leave the application in place, which makes the formal deletion of the application the responsibility of systems administrators, who may have a better idea as to the real need of an application than some end users.

44. Why isn't FastPak available for Java versions prior to 1.4?

Java versions prior to 1.4 had a bug in them that forced GUI programs to terminate by shutting down the JVM via a call to `System.exit()`, which severely impaired the use of FastPak. See the developers manual for details.

45. Your development manuals state that programs must avoid the "System.exit()" call. I don't understand this. I have a ton of development books dating back to Java 1.1 from renowned authors on the subject that say using System.exit() is a normal way to end a program. Please explain.

Your documentation is not incorrect, it's outdated. Exit calls were needed to compensate for a bug in Java releases prior to 1.4. See the developers manual and the Sun documentation for Java versions of 1.4 and greater for details.

46. I find the necessity of having to modify the change from GUI to daemon and back to GUI mode really pretty tedious because I need to open up config file and edit them manually in between. Are you going to change this?

Yes, and it's been done. As of version 1.2.10.55, Update 3, we have a configuration tool in the main FastPak directory available

that will handle this problem.

47. The sales section of your site offers a much lower cost “unsupported” license. Why would anyone buy this?

We feel that the best and most cost effective way for people to purchase our product is to by full licenses for a few people that will become “experts,” as in developers and/or systems administrators, and then buy the lower cost unsupported licenses for all others. This, in our opinion, allows a higher degree of control for a company. It also allows someone to try out our product at a very low price. If you intend to constantly upgrade your programs, Java versions, etc. then this may not be the best pricing option available, although it appears inexpensive on the surface.

