

## PROGRAMMING LANGUAGE

# Java 2 Standard Edition 1.4.2

We haven't looked at Java for some time in *Linux Format*, despite the fact that it continues to march forward in terms of the technology it provides and acceptance among developers. Paul Hudson investigates...

## BUYER INFO

The definitive Java compiler and run-time environment.

- **SUPPLIER** Sun
- **PRICE** Free
- **WEB** <http://java.sun.com>

Java, if I could choose, would be the one language I wished I had more time to study. C, C++, PHP, Perl, and Delphi I use regularly and know like the back of my hand, and yet Java – arguably the most advanced language from the list – barely gets a look-see beyond me trying it out every so often to see what's changed. If your nostrils flared and you got tunnel vision when you read that I thought Java is the most advanced language out there, hold your horses – Java is quite different to both C and C++ because it includes built-in libraries to accomplish most basic tasks (XML parsing, 3D graphics, etc), and that's what I'm referring to.

Although 1.4.2 doesn't sound like a particularly huge jump up from 1.4.1, trust me – there are big changes going on. And if you haven't looked at Java for some time, you should definitely consider giving it a shot.

## Java is too slow!

After years of Java being around, "it's not fast enough" is still probably the biggest complaint. Since it was first released, Java has come a long way in terms of performance, but it's true that still doesn't operate at the speed of native code. Java compiles down to a cross-platform bytecode, that is then interpreted by the virtual machine to perform native instructions. A few versions ago, Sun introduced a hotspot just-in-time JVM, which is to say that the JVM analyses the code it's going to execute, and converts it to native machine code for maximum performance. In this release, however, two things have combined to give Java a substantial speed boost. Firstly, a

The screenshot shows the Java Sun website with a search bar at the top right. The main content area features a heading "J2SE 1.4.2" dated "April 29, 2002". Below this, there are several paragraphs of text, including a welcome message from MDR-Edo and a list of team members: Scott Violet, Stanley Ho, and Ken Russell. The page also includes a sidebar with navigation links such as "Technologies", "Downloads", "Documentation", "Industry News", "Developer Services", and "Java BluePrints".

<http://java.sun.com> is a great repository of resources for all things Java.

special effort was made to focus on performance in the virtual machine, which has resulted in a 30% speed increase in command-line application startup, and 15–20% in small Swing applications. Secondly, 1.4.2-compatible JVMs will hotspot emit SSE and SSE2 instructions if they are supported, which should provide a tremendous speed boost.

## But Java looks awful!

Or perhaps more accurately: "Java looks awful on my platform!" This is probably the second most common complaint about Java – that end-users feel uncomfortable in Java's alien interface. Again, steps have been taken in 1.4.2 to address this, and JVMs now have two new pluggable look and feels (PLAFs) – GTK+ 2.0 and Windows XP. The GTK+ support is more than just adopting a certain look – 1.4.2-compliant JVMs support limited GTK skinning, which means it will attempt to look like the GTK skin you're using system-wide rather than the unskinned GTK default. As you may know, currently GTK skinning support is less than perfect – it only supports three or four skin types. However, better compatibility is one of the targets for 1.5.

The new Windows XP PLAF should come as a relief for developers

deploying to Windows end-users. It works seamlessly, and is only available on XP systems – when you specify the Windows PLAF, the JVM will choose XP if it's running on Windows XP, and the default Windows PLAF on other versions.

In addition, there's a new Metal theme planned for 1.5, which should revive the ailing look and feel – after all, it hasn't changed for some time.

## Java updates too often!

This is the third and last most quoted complaint with Java – that Sun keep releasing new versions, which means people have to continuously download updates and fixes. Again, Sun have taken some good steps to correct this in 1.4.2, and it's a three-pronged assault. Firstly, 1.4.2 includes "Java Update" – a tool to automatically deliver J2RE updates to end users on a regular basis. It's currently only available for Windows, but as its usefulness becomes apparent it's likely to expand to other platforms. The second prong is that J2RE updates are now done using an install-on-demand system, which means that only the features of the J2RE that are required will be downloaded and installed. Finally, 1.4.2 supports incremental downloading – rather than downloading one large chunk of

software, it will instead attempt to download a patch to bring the current version up to date. Three three features combined should result in users being able to keep up to date with the minimum possible effort and download time.

## Conclusion

So, there are quite a few excellent new features for those upgrading from 1.4.1 to 1.4.2. If you haven't looked at Java for some time, then you probably missed the entire 1.4 release – the super-fast new I/O APIs, the excellent XML support built in, the enhanced 2D support, or the Itanium support. Sun are still pushing things forward faster than anyone else out there, which is great to see – I think many developers will be delighted to see features like AES support in 1.4.2, even though the encryption standard was only finalised recently. Users and developers alike will also be happy to see hundreds more bug fixes, both in the compiler and in the virtual machine, to make the platform much more stable.

This is an excellent release, offering a slew of cool new functionality and automatic speed boosts through SSE, SSE2, and the other various performance improvements. It also offers a good taster of what can be expected in 1.5 – better PLAFs (including a new user-definable PLAF that reads its look from XML) and more performance. [LXF](#)

## VERDICT

Features	10/10
Performance	9/10
Ease of use	9/10
Documentation	8/10

Continues Java's tradition of being on the cutting edge of the cutting edge.

**LINUX FORMAT RATING**  
**9/10**