

## Re: Delphi, .NET and Linux

---

*Source:*

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.non-technical/2007-06/msg03443.html>

---

- *From:* "Paul Nichols [TeamB]" <[paul@xxxxxxxx](mailto:paul@xxxxxxxx)>
  - *Date:* Tue, 19 Jun 2007 22:50:00 -0400
- 

Alexandros Karypidis wrote:

Hi all,

I have been assigned the task of evaluating and choosing a RAD tool for the software house I work for. I'm half-way through evaluating PowerBuilder 11 and am about to take on Delphi.

One major concern for us is that our clients are very keen on having the software run on Linux.

If you want the easiest way to do XPlatform, then Java is your choice. You can, of course, write XPlatform apps with other toolsets, but you might as well forget RAD and change it to could be BAD!!

Other toolsets like GTK+ widgetsets, WxWindows will work, but configurations on multiple platforms is painful. Plus, since these widgetsets are not native, you will find stability may be a problem.

Java uses the underlying native calls for whatever system you are targeting. You are writing to the VM, not the widgetset. Give you an example; slots and signals are not the same as Windows loopback listeners and Windows events. The concepts are similar but the implementation is not. Therefore, the widgetsets have to do some trade off. Stability on Windows and Linux with GTK+ or WxWindow apps I have tried (I like to use for instance, GIMP on both Windows and Linux) my reveal that the app runs great on Linux, but will eventually hang and lock on Windows (try GIMP, use it extwnsively, and see what I mean. Gimp is actually one of the best native XPlatform apps I have tried, using X based widgetsets, BTW).

Java, on the other hand, uses a agnostic approach to the underlying OS. It isolates you from the normal way the OS or X layer would be called, by creating their own way of implementation and calling the underlying OS native APIs through the runtime environment layer.

Admittedly, Swing has had traditionally had problems as well with rendering, but the newer versions are much better. Follow the tutorials on threading events and it is not bad at all. My summary on native verses Java and NET can be summarized as follows:

Native:

(1) Native can be faster and will render more closely to precisely to the OS GDI layer employed to the OS/System you are deploying to.

## Re: Delphi, .NET and Linux

However, this may be limited to only when you are using some widgetset that is preferred for the GDI layer you are coding to: (ie) Windows APIs or abstraction layer that calls the APIs directly, like VCL, or like GTK+ on XWindows based systems, such as Linux, Solaris, Mac OS.

(2) There is usually a load time and run time advantages associated with natively compiled binaries.

This is what