

Re: SwingWorker.execute() does nothing

Re: SwingWorker.execute() does nothing

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2008-10/msg00757.html>

- *From:* "Peter Duniho" <NpOeStPeAdM@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 08 Oct 2008 18:47:32 -0700
-

On Wed, 08 Oct 2008 17:47:25 -0700, Ben Phillips <b.phillips@xxxxxxxxxxxxxxxxxxxx> wrote:

Really? What idiot made that boneheaded design decision --

[...]

What /is/ boneheaded is [personal attacks that are not relevant here]

Please do not post unconstructive material, particularly personal attacks.

You would do well to follow that advise as well, being the person who started the "bonehead" theme, as well as calling the person who was involved in the design decision an "idiot".

You're wrong on other counts as well, but when it comes to ad hominem, you're the guiltiest party in this thread.

It is documented under the setVisible() method of Dialog (from which JDialog derives) in the API docs

But that was not where I was looking. Since the problem appeared to be with SwingWorker.execute(), *that* was where I was looking.

You and "zerg/Twisted/Paul/etc." should get along famously. You both have this very strong, unshakable impression that everything should be documented exactly where you look, instead of where you should have looked.

Obviously, documentation of a "gotcha" is not very useful if it is not located where the user is most likely to look for it.

Re: SwingWorker.execute() does nothing

Re: SwingWorker.execute() does nothing

When dealing with the behavior of a Dialog, I would expect the rational user to be most likely to look in the documentation for the Dialog class.

This issue has practically nothing to do with SwingWorker. SwingWorker is involved only because *_you_* brought it into the picture by using the Dialog instance in the same section code as the SwingWorker. You would have had the exact same problem with *_any_* code you'd written after calling the setVisible() method.

Perhaps a note to this effect should be in **both** places, given the likelihood that someone will be showing a modal progress dialog or other such thing during a long-running operation, and given the natural tendency to want to pop up the dialog **first** and **then** start the operation.

That makes no sense. The logical conclusion from your assertion is that every single class method in the Java API should include documentation that says "if placed after a call to Dialog.setVisible(), this method will not execute until after the user has dismissed the Dialog". After all, Dialogs are presented to the user in a practically infinite variety of situations.

No, what makes more sense is that programmers simply should be expected to read the documentation related to the methods they are calling.

Pete

.