

## Re: Giving an application a window icon in a sensible way

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2006-11/msg02677.html>

---

- *From:* "Twisted" <[twisted0n3@xxxxxxxx](mailto:twisted0n3@xxxxxxxx)>
  - *Date:* 21 Nov 2006 15:42:52 -0800
- 

Daniel Pitts wrote:

I see quite a few links to `ClassLoader.getResource`, hmm, how about that.

For a query of "java resource loading", not "java icon loading" which is one of the ones I used, and other similar ones.

It is not exactly fair to fault me for not using the exact query you just used. Also, there's the factor that you know exactly what answer you're looking for, so a) when you tried one related query and didn't get it, you tried more queries and posted one that produced the planned conclusion, and b) you recognized the significance of "`ClassLoader.getResource`" when you saw it in excerpts by search hits, whereas I went by the text in the excerpts in deciding what looked like a promising result.

This supports my conclusion that one frequent cause of problems is that search engines work mainly when you already know, in detail, exactly what you're looking for. It's easy to find Porsches with it; hard to find a car recommendation for a given requirement and budget. Easy to find `ClassLoader.getResource`, hard to find "how to incorporate application icons portably in Java". Etc.

Search engines then end up good for finding references for stuff you already know, or detailed information on something you vaguely remember but already had encountered; much harder for answering questions where you don't already have a good idea of (at least part of) the answer. In fact, there seem to be three levels of difficulty:

- \* Digging up references to something you already know, or the `faq/manual/whatever` for something you already use: easy. Some questions you're unsure of too -- was xyz play by Shakespeare? What else did he write? Where are certain places, cities and such? Who first said <quotation>? If you have an exact name for something, and it's THE name rather than one of many choices, it's definitely easier.
- \* More difficult: answering questions you can state in plain English,

## Re: Giving an application a window icon in a sensible way

but for which pretty much every key word has many synonyms and the proper names involved (in our example, just Java) are not sufficient to narrow things down much ("Java" by itself still leaves a huge body of subject matter). This can especially be the case if you have a specific problem to solve and phrase your query in its terms, and the ideal #1 search result solves a general class of similar problems in more general terms, resulting in a much shoddier result ranking. In my case, the answer you think I should have gotten probably was ranked #3,000,000 or so. Even if it was ranked #3, the text excerpt mustn't have seemed as relevant as for one of the other top ten. And of course if there are two solutions of which one is (arguably) superior, it's still the one that someone finds \*first\* that gets implemented, as a rule. :)

\* Most difficult: anything whose "query" isn't even translatable into words. The ones that seem to come up the most are "put a name to this place/face" and "find pictures like this". The latter has come some distance in recent times, thanks to large image databases at Google and elsewhere; \*if\* you can describe what you're looking for in words and \*if\* a picture has been put on the web somewhere with those same words in close proximity, you stand a chance. Finding more pictures of something you can hang an unambiguous proper name (rather than a description) on works best, which often reduces to the first of the two items I mentioned: putting a name to something you've seen that was not, however, conveniently labeled with it.

Now search engine technology is one of my interests, so I've even recently put these last cases to a test of sorts, or recent search engines to a test on them, mainly Google Image Search. I had some photographs of unidentified skyscrapers and architecture, and also some of celebrities, for some of which I had good guesses and for others of which I didn't. The former, when the guess was accurate, were fairly easy to confirm; the latter proved well-nigh impossible. For the former, the guess would turn into a query and a search; if the image or one very like it cropped up in the results it was likely the guess was correct, and more so if there were multiple hits on that same image. For the latter, random stab guessing sometimes worked. I found that it was easier to nail down architecture in that case — queries for "famous skyscraper" and similar turned up a hit within a page or two for many, and the link led to an identification that could be confirmed from a couple other sources easily thereafter. Celebrities on the other hand appear to be a dime a dozen, whatever salaries they command from Hollywood and elsewhere. It's even worse than that — rather than a query analogous to "famous skyscraper" (say "blonde actress") turning up thousands of hits with the first occurrence of a particular one hundreds of hits down from the top, such queries actually turn up a fairly limited selection that can miss some entirely, because where the images occur they are described in more specific terms, usually with a proper name. In fact, to ID unknown celebrities it is more useful to put a description in a text search, and generate guesses that way, then plug those into an image search.

## Re: Giving an application a window icon in a sensible way

Of course, I've heard that face recognition is coming soon to a search engine near you, although eventually we need tools that can more generally parse images to generate some keywords (for example, by reducing an image to a 16-color VGA palette and pixel-counting you can compute reasonable weightings against keywords like "red", "green", "dark", and so forth; not that any of the major image search engines appear to do even that much). And, of course, until then putting a paragraph-length description next to every image on the internet would help tremendously -- not only for search engines, but in providing an alternative for visually impaired humans in the form of a description they can try to visualize, much the way novels without illustrations have to present their characters and settings. And of course, this is getting off-topic...

Search difficulty category number two is what is of more concern right here and now, namely, being able to find a general result from a more-specific query, and being able to find something you can describe but can't name (which would go a long way towards helping in area #3, too, for that matter). Google has some features that help here, for instance in that it seems to conflate variant spellings and detect possible typos. Build in a thesaurus and grammar awareness and you might be getting somewhere -- for example, homographs such as "lead" (the heavy toxic metal) and "lead" (as in the blind leading the blind) can be distinguished by their part of speech (one is a noun and one is a verb in this example), and synonyms can be fuzzy-matched (so that for a query of "red" the word "red" gets a strong hit, but "maroon" or "mauve" counts for something rather than nothing; "automobile" and "car" may be treated almost identically, with fractional preference being given to hits that use the exact word that's in the query instead of the other one).

Was this in a forum? Just because its on suns site doesn't mean its reliable.

It looked like a developer network article rather than a joe random forum posting to me. Also, considering that \*it bloody worked\* (with a little tweaking) it doesn't seem to have been "unreliable" by most sane definitions of the term.

Actually, most of them suggested using `ClassLoader.getResource`, and that if you thought it was difficult to create jar files, try using ant instead. The suggestion may have been off topic, but it was only meant to help.

I'm sure that when it comes time to distribute something, creating jar files won't prove too difficult. But I'm leery of creating something some of whose functionality is dependent on being in a deployment

Re: Giving an application a window icon in a sensible way

## Re: Giving an application a window icon in a sensible way

rather than a development environment. Understandably so! Having to bundle, install, test, debug, edit rather than just test, debug, edit would slow down the testing/tweaking/improvement cycle drastically. And would it even be possible to run it from a jar yet attach a debugger? If not, then there could be functionality and code paths that could only run without the debugger attached, making the debugging of those code paths a history refresher on Victorian software development methodologies. Not exactly what I am looking for in the way of continuing education right now.

5. The very last place to go for trustworthy information on doing stuff in Java is [www.sun.com](http://www.sun.com). Google and wikipedia must also be stringently avoided.

This seems to be what you do, I usually get my information from Google, or perhaps a book written on the subject.

Somebody obviously hasn't been reading this thread. Which is fine, except that the same somebody nonetheless posted an opinion into the thread in question, and a strong one at that.

In the first few posts it becomes apparent that I made some Google searches that turned up nothing that \*looked\* relevant (to someone who didn't already know what they were looking for, anyway -- and those are the someones good search results are most needed by), then posted here, then after a while passed without responses tried some more Google searches and eventually got a hit that led to a method \*that worked\*.

In other words, I ended up solving my problem with Google, contrary to what you suggest, and the suggestion that I solved it "wrong" is therefore a suggestion that I shouldn't have used Google, given what the result was when I did, and that that result is supposedly "wrong" (despite having actually worked).

As for "books written on the subject", that is an option for someone with a higher budget than I. Keep your recommendations and (revenue-generating, no doubt) Amazon links to yourself, please. And don't you \*dare\* suggest that "if you can't afford xyz, you shouldn't touch any development tool with a ten foot pole", lest I call you a filthy capitalist pig that discriminates against the poor and supports raising the barrier to entry to entrepreneurial activities to protect an incumbent CEO class from any risk of ever facing something resembling actual competition, then launch into a lengthy political diatribe. (DISCLAIMER: I am not a communist. I just oppose fiscal policies that seem designed to raise artificial barriers in the face of the lower classes to keep them from ever climbing up and threatening to unseat the incumbents. In other words, I am better described as capitalist but anti-fascist and somewhat libertarian-leaning.)

Re: Giving an application a window icon in a sensible way

## Re: Giving an application a window icon in a sensible way

Actually, it is important to understand standards and to adhere to common conventions and best practices.

Even ones that are documented either a) nowhere or at least b) nowhere you can find them if you don't already know them? Curious.

Actually, I think you've come up with a complicated solution.  
Step 1, convert image to XPM,  
Step 2, run some tool to convert it to a class file.  
Step 3, move the class file into the correct place.  
Step 4, use some uncommon third party tool to load and convert the class's data..  
Step 5, Fix the third party tool, because it was broken.  
Step 6, Fix the third party tool again, because it was broken in another way.

All the complication in question is up-front, at development time. Once solved, it stays solved, and there's no runtime complication involving I/O that can fail. If the binaries loaded, it will work. If they didn't, then there are bigger problems anyway.

The way that most of us have suggested takes this form:  
Step 1, move or copy your image into a directory in your project

What directory? How do I make it work seamlessly whether it's in the development environment or the deployment one? None of that has yet been explained to my satisfaction; all I've seen are vague assurances that this can be done, without any detail.

This directory is apparently to be on the class path, which suggests one of two consequences. Either every project winds up adding another dir to the class path, or all the files from every project end up in a single dir jumbled together (and then when one project is packaged, you risk including extra files that waste disk space or, in trying to avoid that, accidentally omitting a file that is actually needed).

If you're not going to take the advice of people on comp.lang.java.\*

I didn't say I wouldn't. I did say I wouldn't follow it *\*blindly\**, and certainly that I won't follow "advice" that consists of vague suggestions lacking detail in crucial areas. "Get and use tool X" without any real detail (not so much as the URL where the tool's

## Re: Giving an application a window icon in a sensible way

official Web page resides, assuming it even has one) and "Do this" suggestions that leave a lot of questions that, when asked, go unanswered except with flamage, do not exactly encourage me to be trusting.

I still haven't heard anyone tell me any of the following, which is behavior that I find suspicious:

- \* From those who recommended getting Ant, its official URL at minimum.
- \* Regarding getResource, the exact way to have the resource found when the app is tested in the development environment \*without\* either jumbling every project together \*or\* overloading the system class path with a sub directory for every project.
- \* Regarding getResource, the exact way to have that same resource automatically included into a jar when one is built, when the time comes. One person suggested that Eclipse can be made to do this, without saying anything in detail about how. Others suggested Ant would be needed, without saying anything in detail about how to do it with Ant.

Unfortunately, I have no confidence that I can answer those questions to your satisfaction purely by my own research and wits. I am somewhat more confident that I can answer them to \*my\* satisfaction that way, mind you; but the events of this thread make it clear that even if I do, the method I come up with via my own research efforts might nonetheless be decried as "wrong" and lambasted with what seems to be almost \*religious\* fervor...

My suggestion to you is that if you want me to answer those questions to your satisfaction (rather than just to mine) then you should give me the answers you want me to end up coming up with. Or is this your standard MO — tell someone nothing until they try to solve the problem themselves, and then criticise them like crazy for all of the (perceived) shortcomings of their solution?

.