

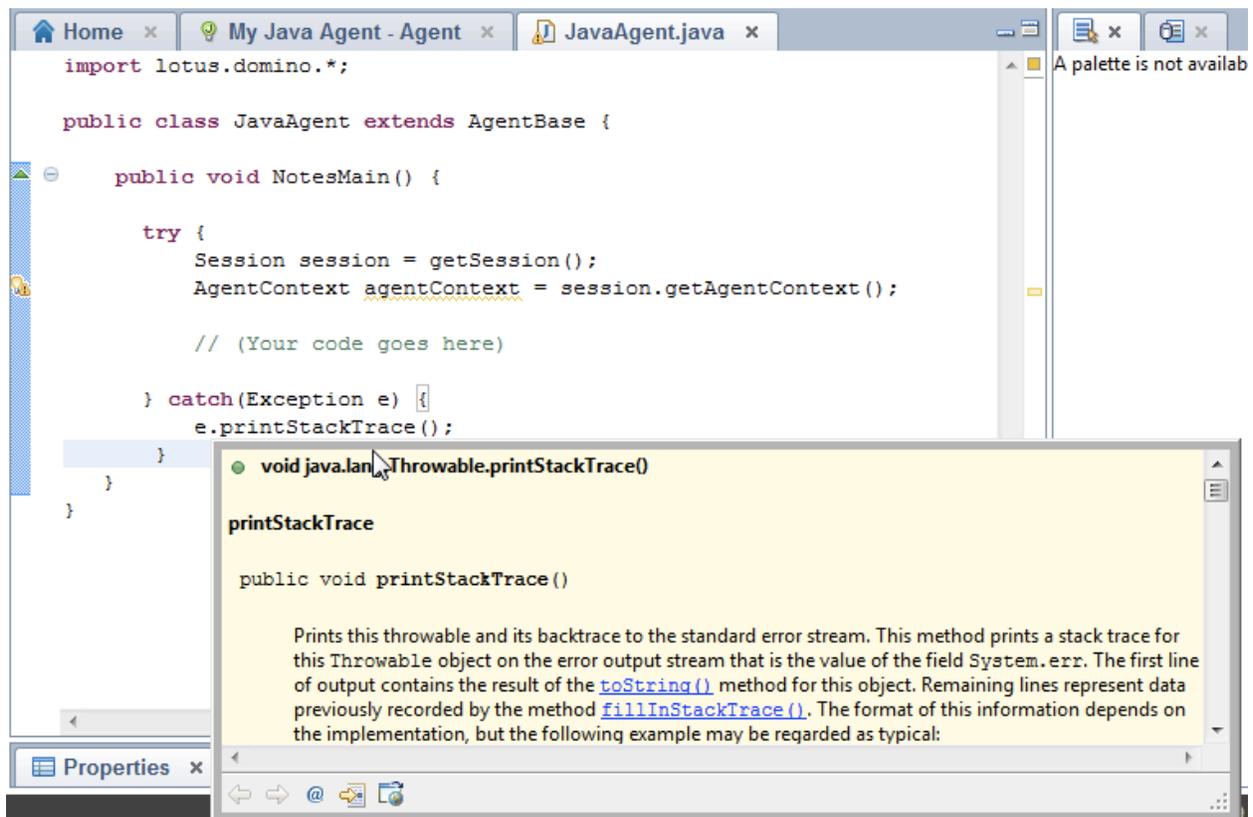
# Adding Hover Help to Domino Designer

by Julian Robichaux, panagenda

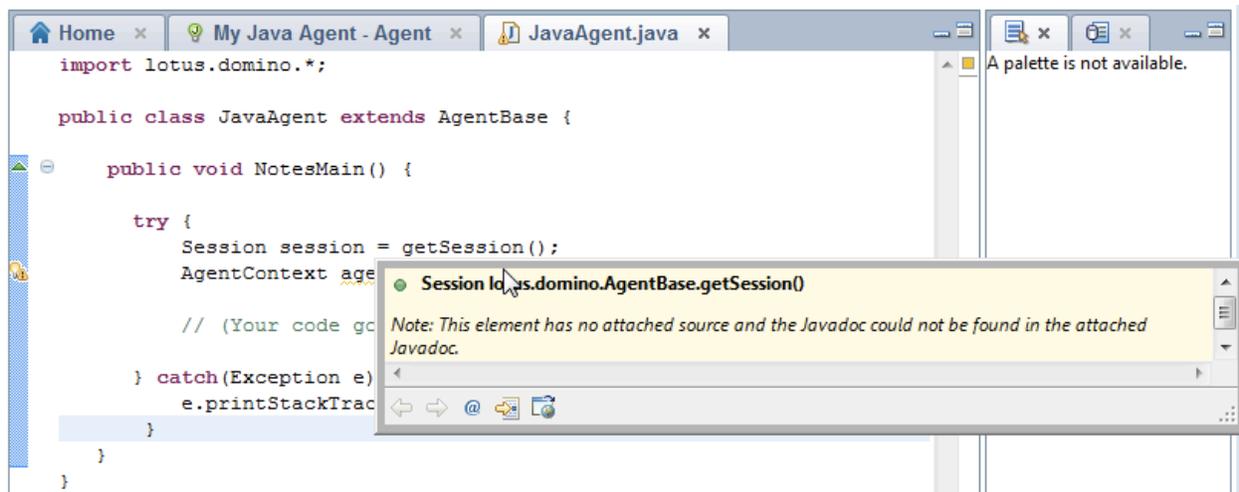
originally published on [socialbizug.org](http://socialbizug.org), May 2013

When IBM migrated Domino Designer to run on Eclipse (DDE), we automatically gained a lot of very nice functionality for Java programming from the Eclipse IDE: refactoring, content assist, auto-imports, code folding, and much much more.

One of the pieces that was never fully realized, though, was hover help. If you hover over a native Java method and wait for a few seconds, you'll see something like this:



However, if you hover over the method of a Domino class you'll get a message saying "the Javadoc could not be found":



But wait, you can fix that!

You can tell Eclipse (and therefore DDE) where to look for Javadocs. However, since IBM has never (and still does not) produce Javadocs for the Domino java classes, and because we also do not have the source code for these classes, we can't just run the Javadoc tool against Notes.jar and generate what we need. There are other tricks involved.

Enter the Domino Javadoc Generator. This is a Sourceforge project created by Ferdinand Prantl way back in 2006 to address the very issue of missing Domino Javadocs. It is an ANT script and a Java utility that work together to do the following:

- extract a list of Domino Java classes and methods
- extract the descriptions of classes and methods from Domino Designer help
- combine those things together into a proper Javadoc file

Here is how you integrate that into DDE.

### Step 1

Go to <http://domino-javadoc.sourceforge.net> and download domino-javadoc-0.4-out.zip from the main page (this file is referenced and linked to several times on the page). It is a pre-built set of Javadocs, so you don't have to go through the extraction process and ANT script setup yourself.

### Step 2

Inside the zip file, there is a file called Notes7-pseudo-javadoc.zip. Extract that file to a path on your hard drive that is permanent. In other words, don't put it in your temp directory where it might get deleted. The Notes jvm folder was a logical choice for me.

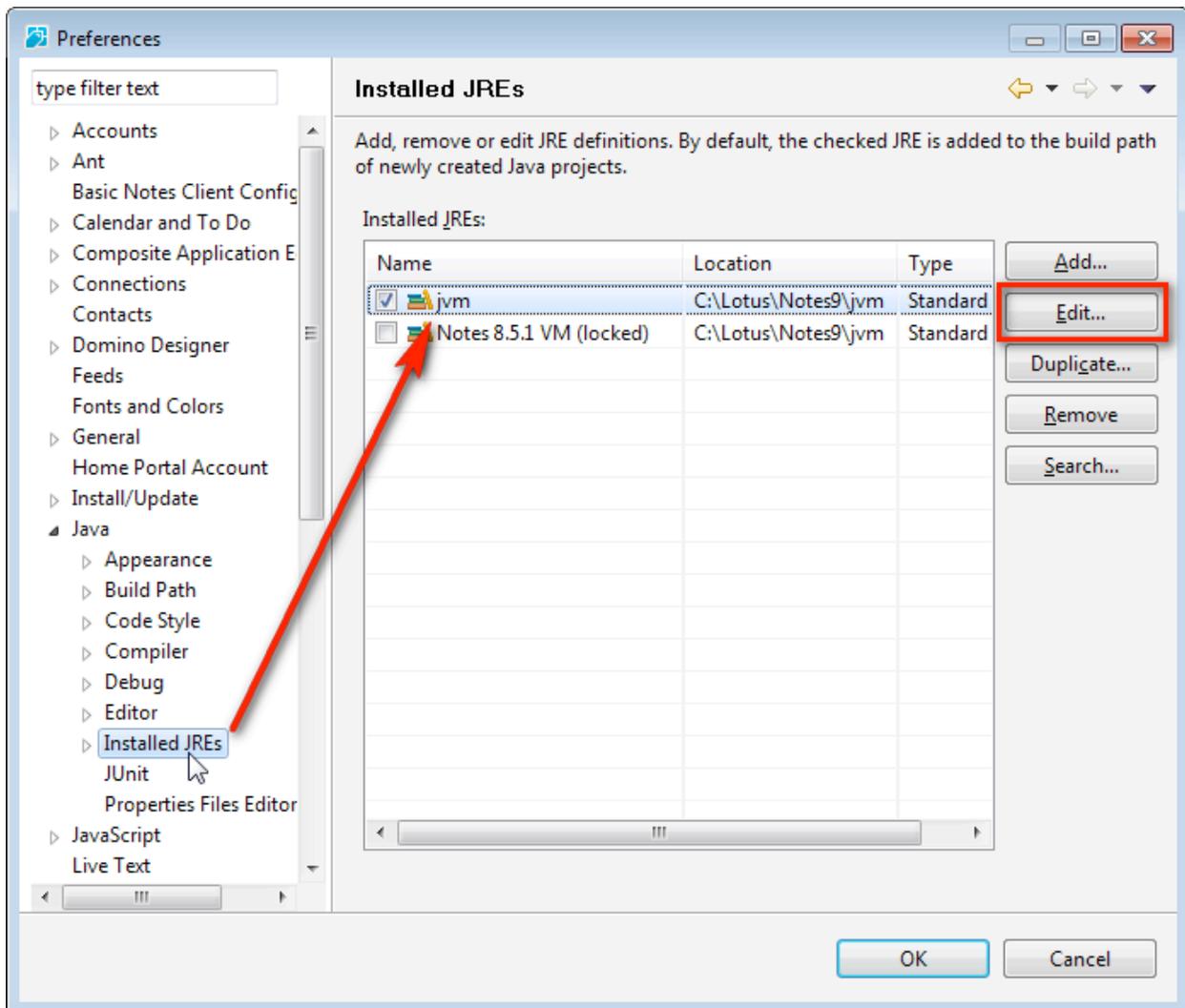
### Step 3

Open DDE and:

choose the menu option File - Preferences

navigate to Java - Installed JRE's in the left-hand panel

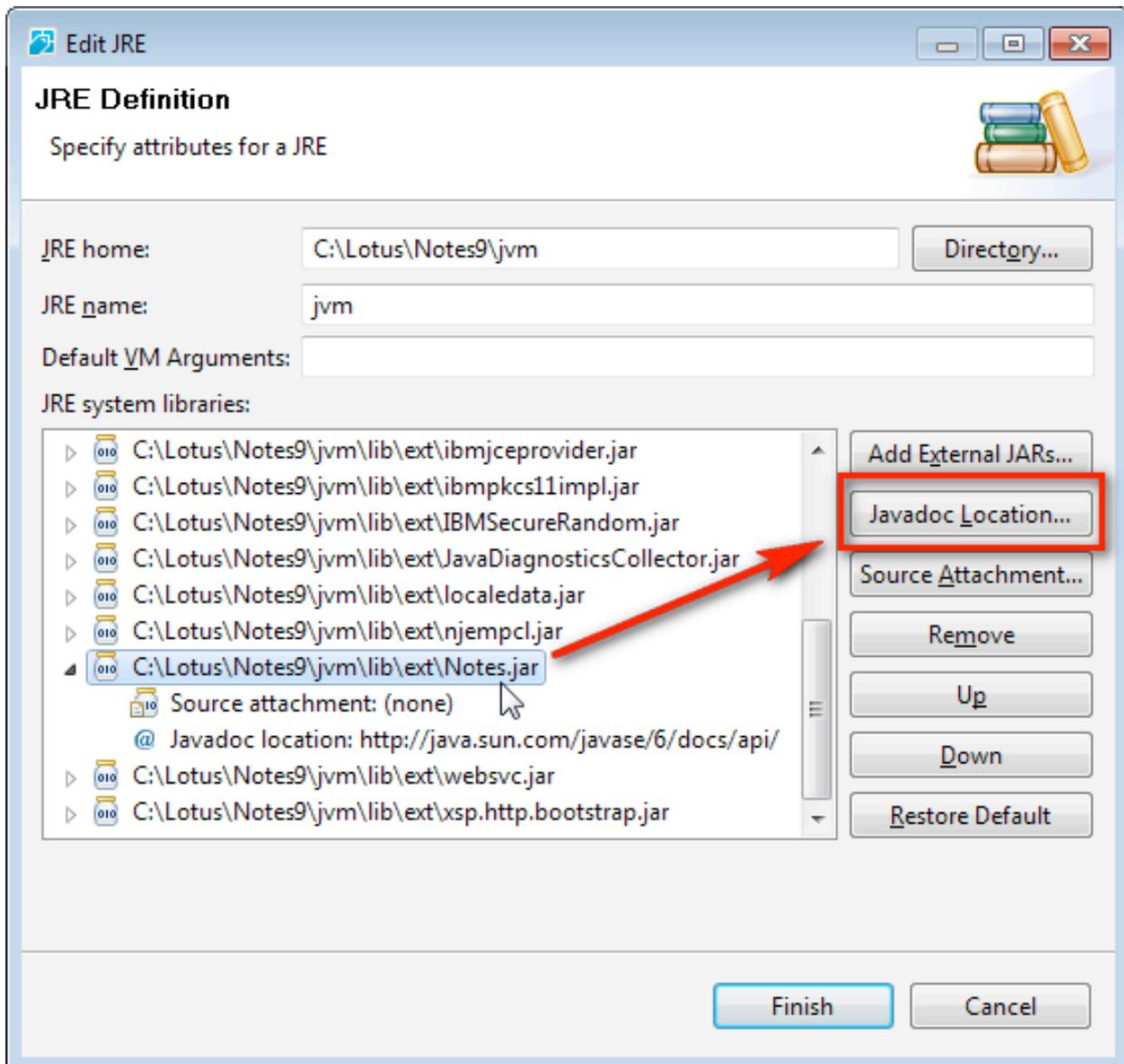
highlight the "jvm" JRE and click "Edit"



### Step 4

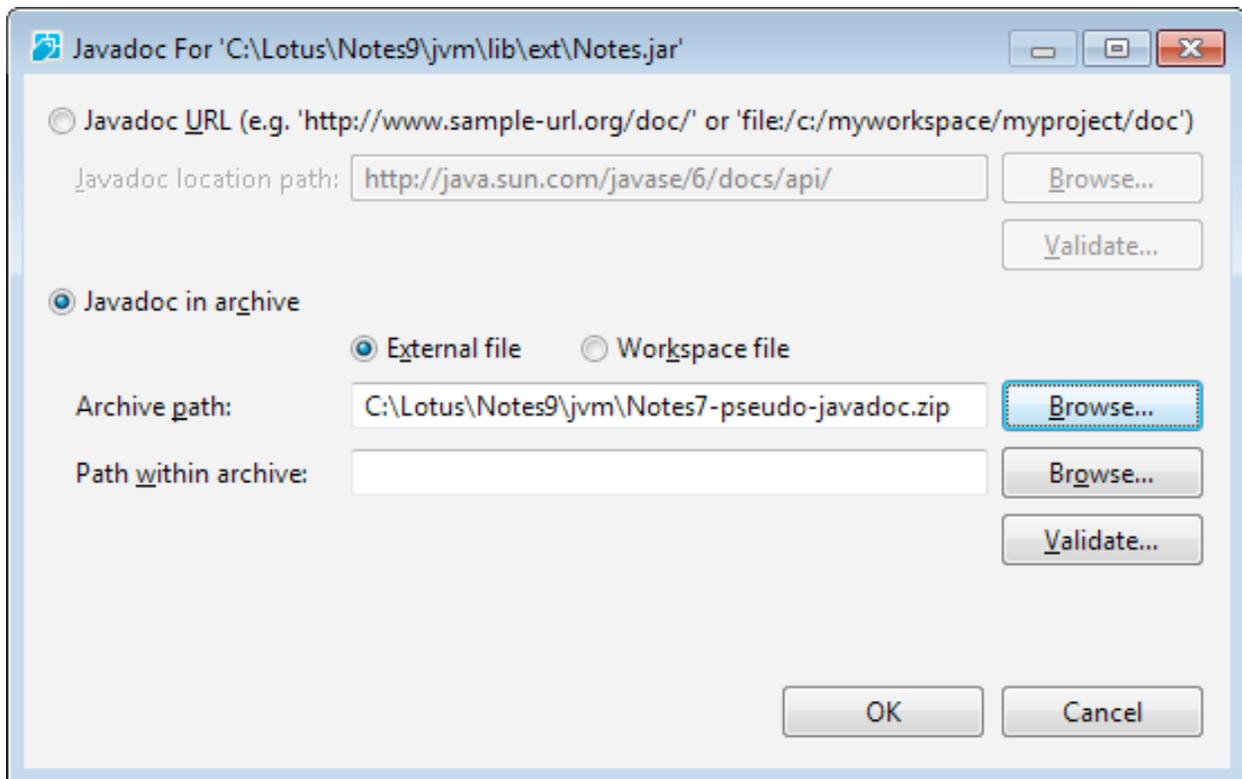
In the "JRE system libraries" list, scroll down to Notes.jar and select it.

If you expand the Notes.jar entry, you will see that the current Javadoc Location is "<http://java.sun.com/javase/6/docs/api>". This is obviously incorrect.



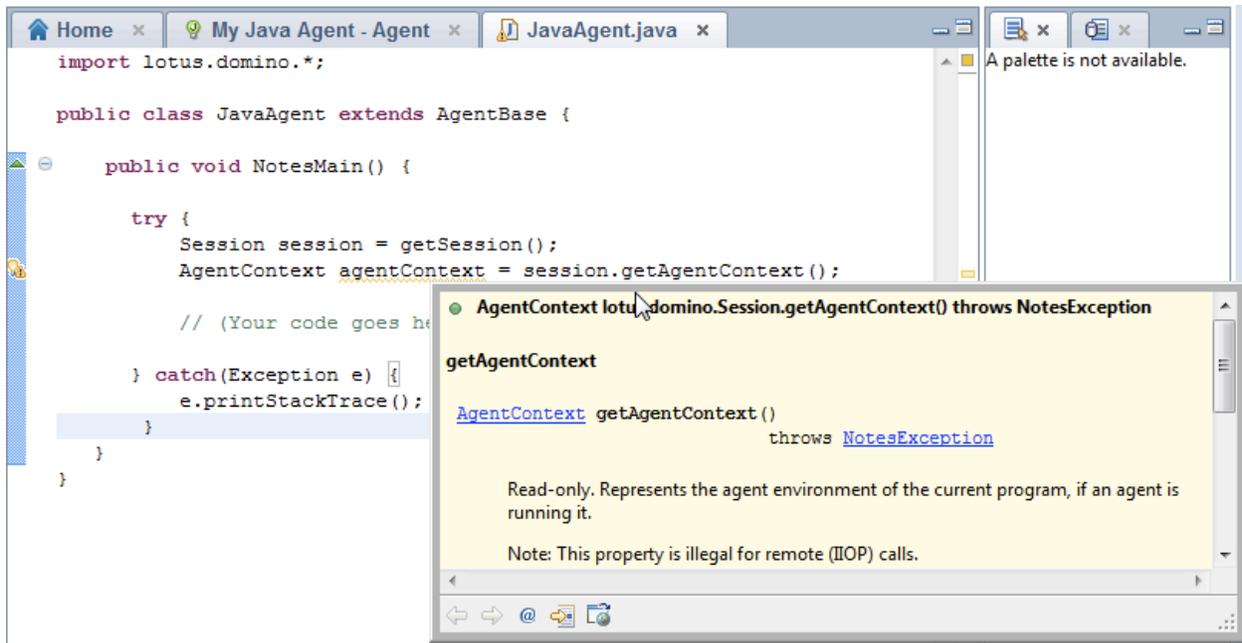
### Step 5

With Notes.jar selected, click the “Javadoc Location...” button. In the dialog that appears, choose “Javadoc in archive” and use the “Browse” button to indicate where the Notes7-pseudo-javadoc.zip file is located (this should go in the “Archive path” text box).



Then click “OK” and “Finish” as many times as it takes to save your changes and back all the way out to DDE.

Now your hover help in DDE should be much more... helpful.



### Notes on this process:

I am certainly not the first one to have done this, so I am not trying to take credit for a new discovery. I'm just giving you the steps involved. A search for "eclipse add javadoc" will produce similar steps, and at least one Domino blogger has also written about this (<http://devblog.virtage.com/2013/03/add-missing-javadoc-to-lotus-domino-designer-help>).

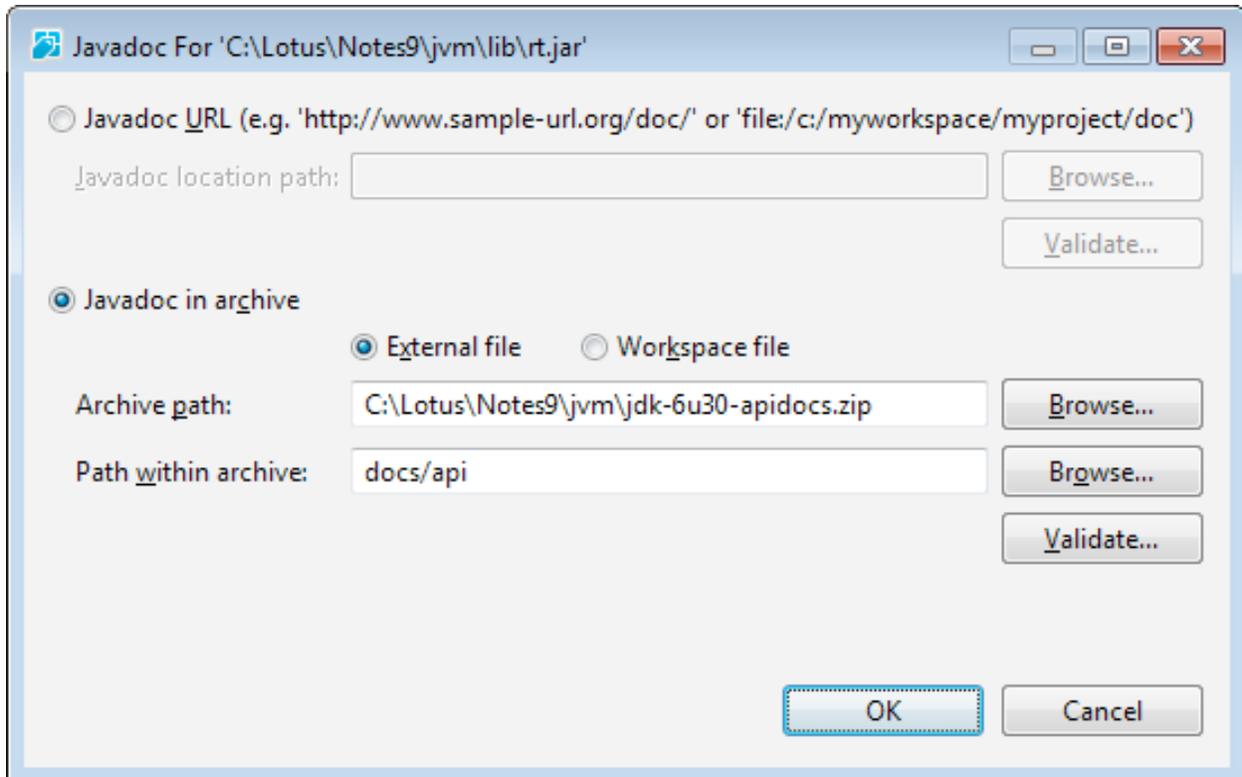
If you get error messages about the Javadoc package not being valid, check two things: (1) make sure you are pointing to Notes7-pseudo-javadoc.zip and not domino-javadoc-0.4-out.zip, and (2) try putting the javadoc.zip file on a path with no spaces.

The Domino Javadocs we downloaded and used in this article were generated in Notes 7. Feel free to follow the instructions on the Sourceforge website to generate a brand new batch of Javadocs from Notes 9. Between Notes 7 and Notes 9 I think there are only about 30 or so new Java methods (I am sure someone will correct me if I'm wrong), so for me the Notes 7 docs were close enough.

### Addendum

While you are in the process of adding Javadoc locations in DDE, you might want to consider downloading the Java SE Javadocs from the Oracle website (<http://www.oracle.com/technetwork/java/javase/downloads/index.html#docs>) and adjusting the lookup for vm.jar and rt.jar to point to a local Javadoc file instead of a URL. This will make the hover help for standard Java classes work much faster because it will search locally rather than accessing the Oracle website over and over.

Be aware, though, that the way the Oracle Javadocs are zipped up, you will need to add “docs/api” in the “Path within archive” text box right below where you specify the location of the Javadoc file (make sure you use a “/” and not a “\”).



Also, it might make sense to extract the Javadocs from the zip file first and point directly to the unzipped Javadocs files instead of a zip file. It will take up a lot more space on your hard drive (over 200mb), but it will also prevent DDE from having to use extra memory to navigate the large Javadoc zip file each time it looks for hover help. If you do that, you will need to use the “Javadoc URL” option rather than “Javadoc in archive”.