

Java Scrapbook Pages

by Julian Robichaux, panagenda

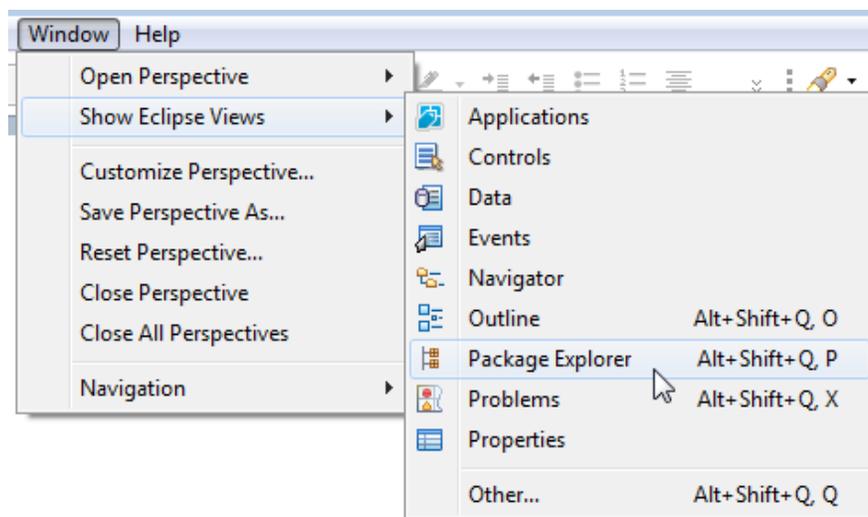
originally published on socialbizug.org, September 2013

Testing small bits of code in Domino Designer (DDE) can be tedious. For most people, it involves writing a small agent, compiling it, saving it, running/debugging it, and iterating through that cycle until you're satisfied with the results. And then, of course, you delete the agent if you remember to (or you end up with a bunch of Test1, Test2, etc. agents strewn throughout your Agents view).

If you're a Java programmer, an easy shortcut in this case is Java Scrapbook Pages [http://help.eclipse.org/juno/topic/org.eclipse.jdt.doc.user/reference/views/ref-view-scrapbook.htm?cp=1_4_7_8]. These are like code elements (Eclipse views, actually) that you can use to quickly run and test chunks of code. You can even do some basic debugging/inspecting on the code. Here's how they work.

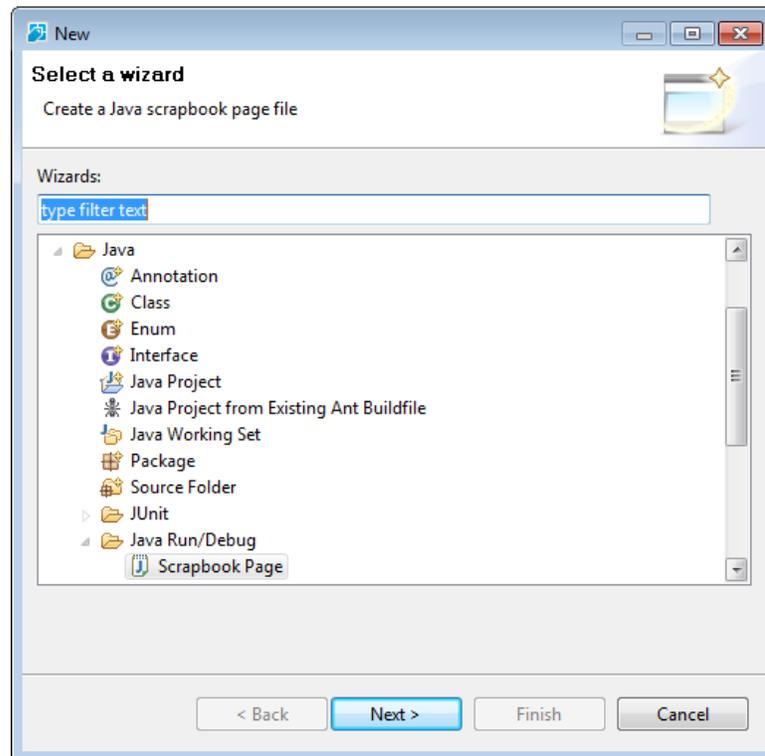
Step 1: Switch to the Package Explorer View

If you don't already have the Package Explorer view active in DDE somewhere, use the menu option Window - Show Eclipse Views - Package Explorer to open it. (If you don't see it on the main list of Eclipse views, you can choose "Other" and then General - Project Explorer).

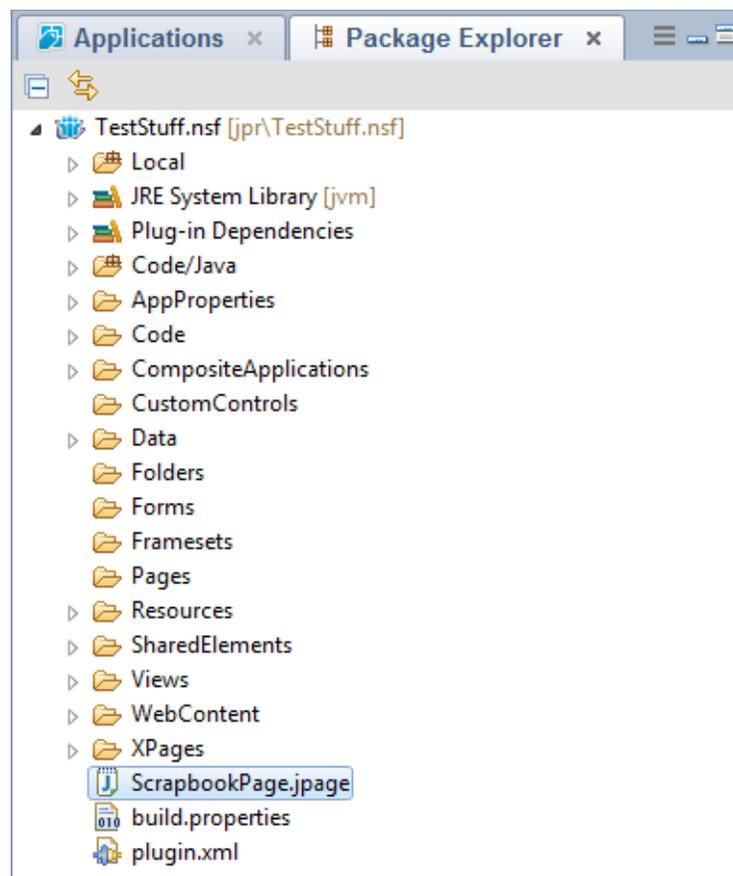


Step 2: Right-Click the Database to Create a New Scrapbook Page

In the Package Explorer view, right-click the database (application) you want to work with and choose the menu option New - Other. Then go to the Java - Java Run/Debug section and select "Scrapbook Page".

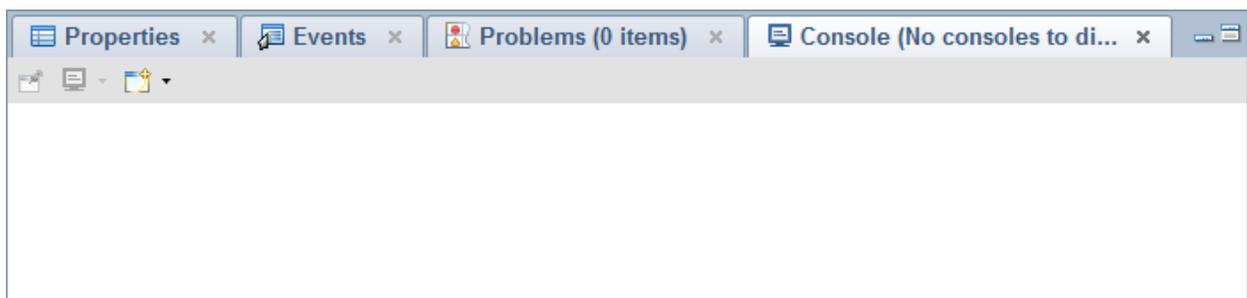
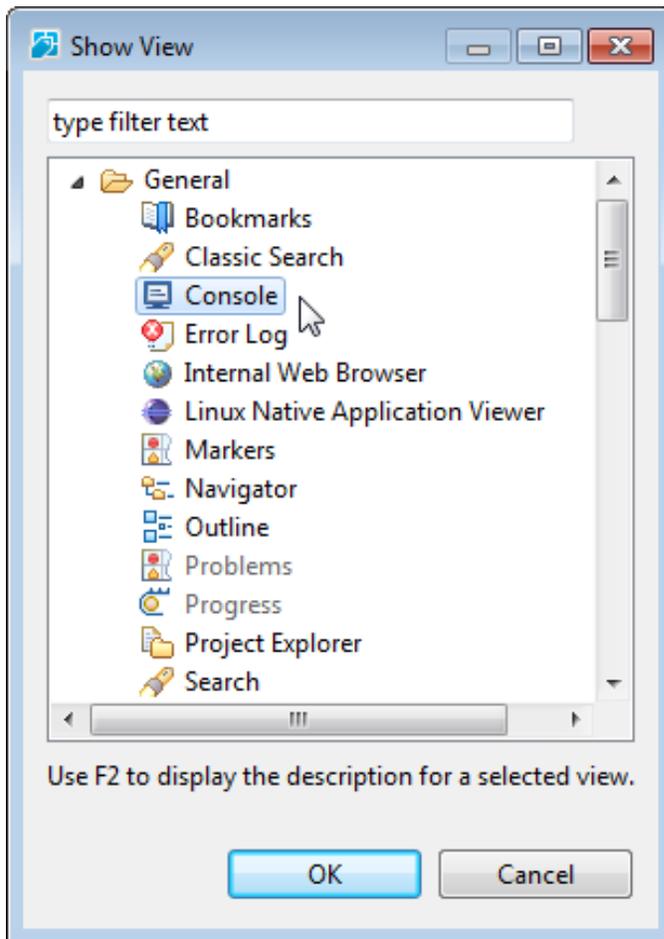


Click “Next”, and then give it a meaningful name like “ScrapbookPage” or “Scrappy”, and then click “Finish”. You will now see YourScrapbookPage.jpage listed at the bottom of the list of files and folders for your application.



Step 3: Open the Console View

You will also want a Console view open in DDE, so you can see the result of System.out calls. To do this, use the menu option Window - Show Eclipse Views - Other - General - Console.

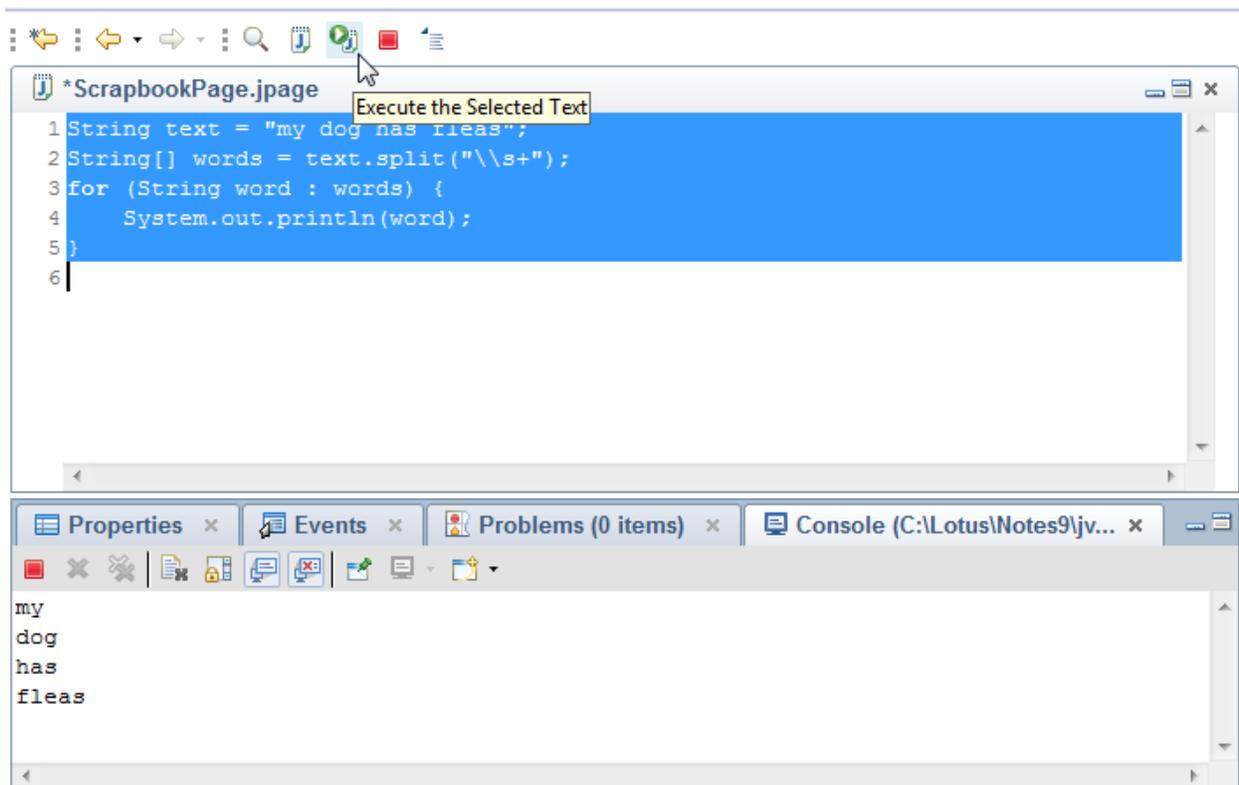


Step 4: Evaluate Some Code

Now you are ready to write and test some code. Double-click your Scrapbook page in Package Explorer to open it up, and enter the following into the empty page:

```
String text = "my dog has fleas";
String[] words = text.split("\\s+");
for (String word : words) {
    System.out.println(word);
}
```

Now select all the code (either with your mouse or using Control-A) and click the “Execute the selected text” button in the toolbar - this is a button with a picture of a notebook page with a “J” on it, and a green “play” arrow. This should result in the following Console result:



As you can see, everything you print to System.out.console will show up in the Console view.

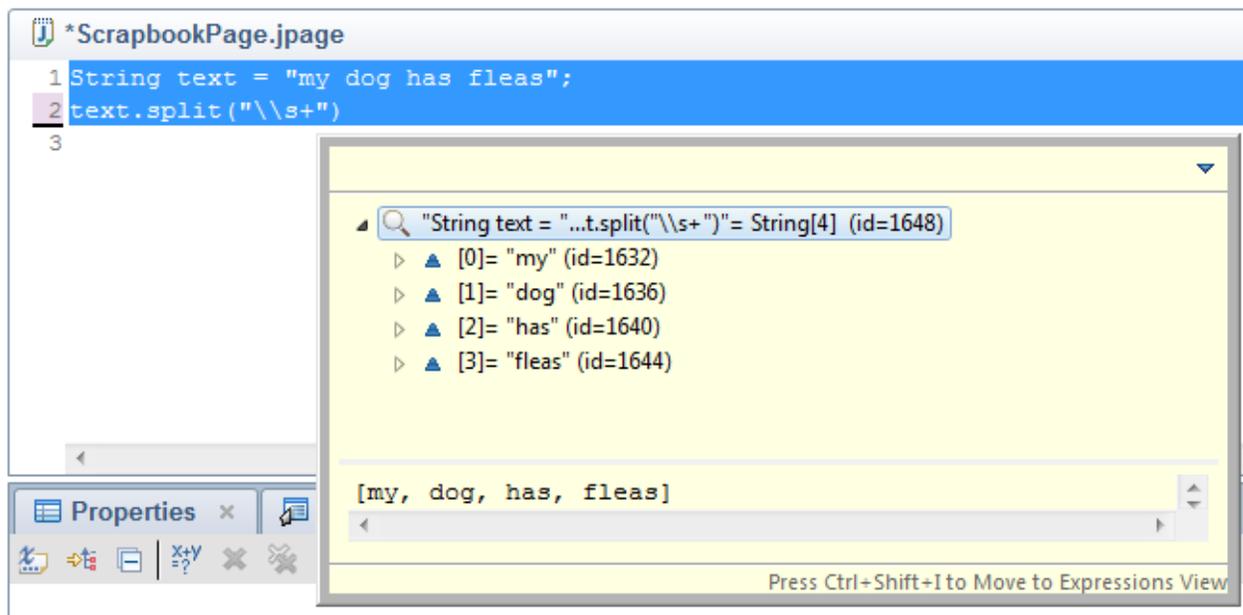
When you're done, don't forget to click the “Stop the evaluation” toolbar button, which is the red square next to the Execute button.

Step 5: Inspect Some Code

You can also use the “Inspect” toolbar button - the one that looks like a magnifying glass - to do basic debugging. This isn’t a full debugger by any means, so you can’t set breakpoints or modify values on the fly or anything, but the Inspector will give you the structure of the last object that was created in the Scrapbook page code. For example, change the code on your page to this:

```
String text = "my dog has fleas";  
text.split("\\s+");
```

No System.out statements, no final variable assignments. At the end of this code we have created a String[] object that can be inspected. To see how this works, highlight all the code again (mouse or Control-A) and click the “Inspect result of evaluating selected text” toolbar button. You will get something like this:

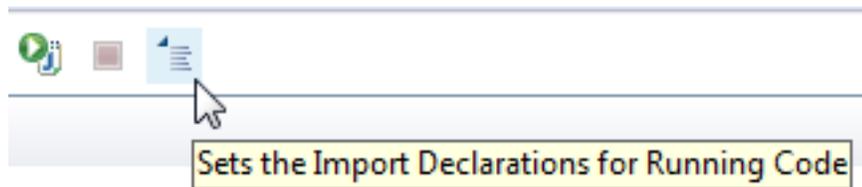


This gives you a small Inspector window you can use to drill down into the contents of the last Java object we created; in this case, a String[] result of the split() method.

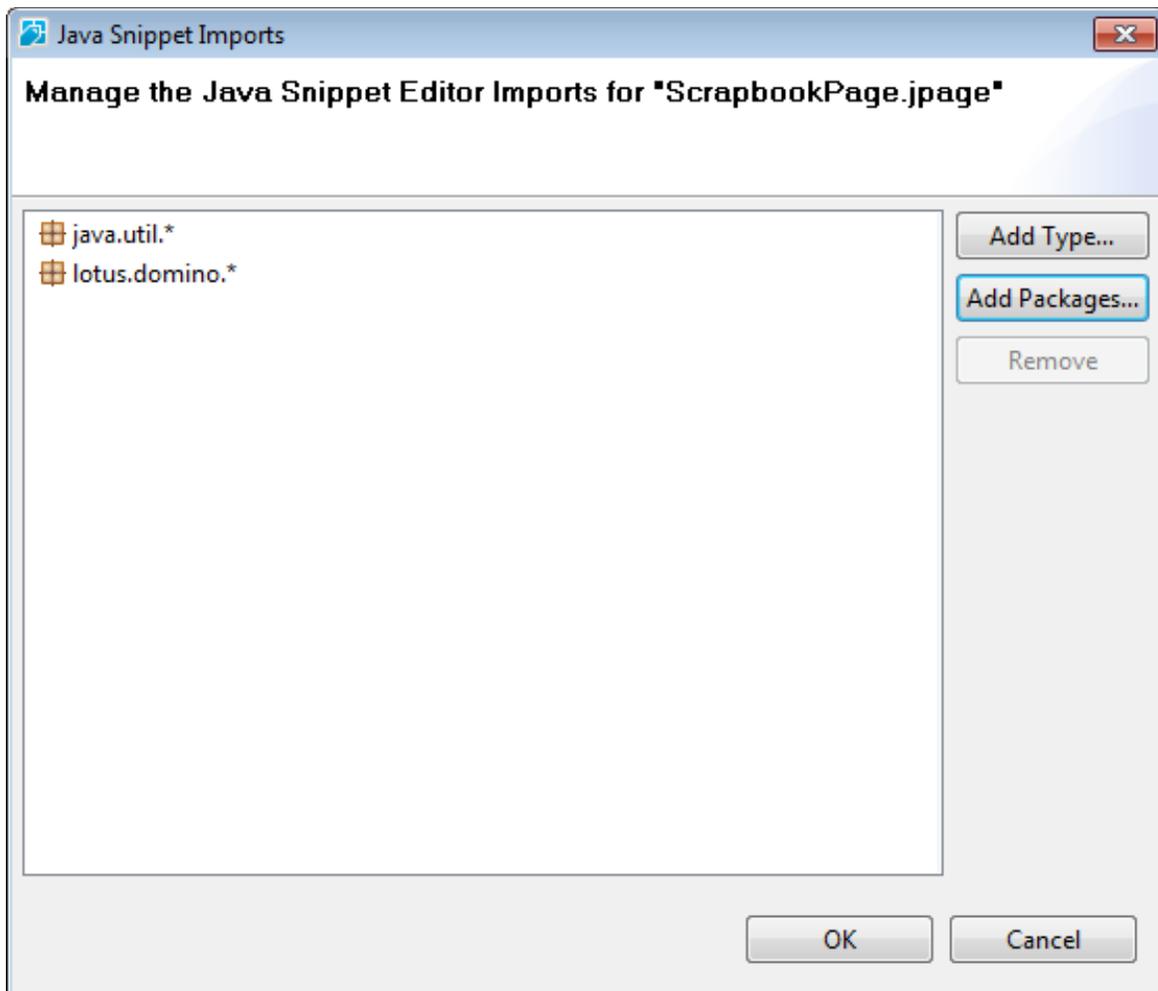
Adding Imports

One thing you will notice very quickly is that you can’t use import statements, and you have to use the fully qualified class name of any class that’s not in the java.lang package.

To get past this, use the “Set Imports” toolbar button.



This will open a dialog where you can add packages that should be auto-imported, like `java.util.*` and `lotus.domino.*`.



After you add imports, the classes in the packages you specify will be available without having to use fully qualified names, and type-ahead will work properly for those classes.

Examples of Use

While you wouldn't want to test large or complex pieces of code here, Scrapbook pages can be extremely convenient when testing things like:

- String evaluations and manipulations
- Regular expressions
- XML parsing
- Numeric calculations

This will never replace a full debugging environment, but it might save you some time and frustration.