Making Domino Designer work like you want

If there's any common refrain I hear from developers when it comes to Domino 8.5, it's that they have issues with Designer. Struggling with an IDE is always painful, no matter the platform. Simple issues like how to auto-format code or how to control content suggestions can make a developer feel incompetent and frustrated in about 10 seconds. Developers are, in fact, the worst kind of users -- always expecting their tools to implement the "do what I want" interface.

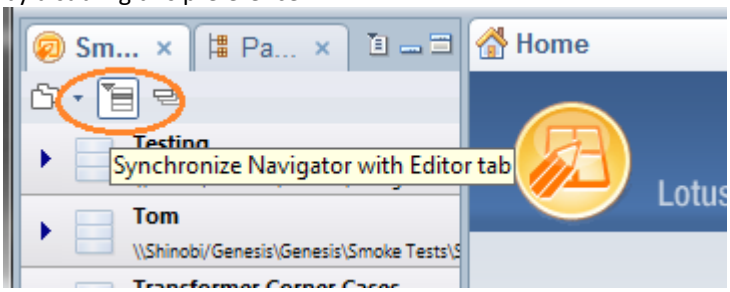
With great power comes great responsibility, and the fact that Domino Designer 8.5 is based on Eclipse means that developers have an enormous amount of power over the behavior of the toolkit. I find that even the most seasoned developers haven't really explored this power, working pretty much with the default settings out of the box and never looking very far past the surface.

I have the good fortune of having a financial motivation when it comes to Domino developer productivity, and so it makes sense for me to spend time on finding the right settings for Designer that make it an incredibly productive platform. With the launch of our partner program with the GBS Transformer, there is even more demand among our partners to get the most out of Designer.

So I decided to put together a guide for the Lotus community on how to get the most out of Designer. In some cases, this means telling Eclipse to DO LESS because what its doing doesn't really help, and just eats up clock cycles uselessly.

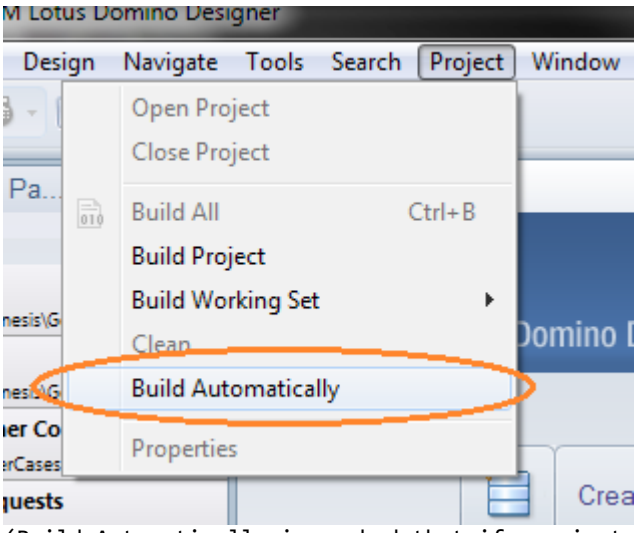
The first step is to keep Designer from being spastic and doing weird unnecessary things all the time. Fortunately, this can be accomplished with a few easy changes.

First is the Application Navigator. The very first thing you should do is make sure that Editor synchronization is turned OFF by disabling this preference.




```
(Don't change this if you like the Navigator to act like a ferret on crystal meth. Some people are
into complete context shifts without warning. Just sayin')
```

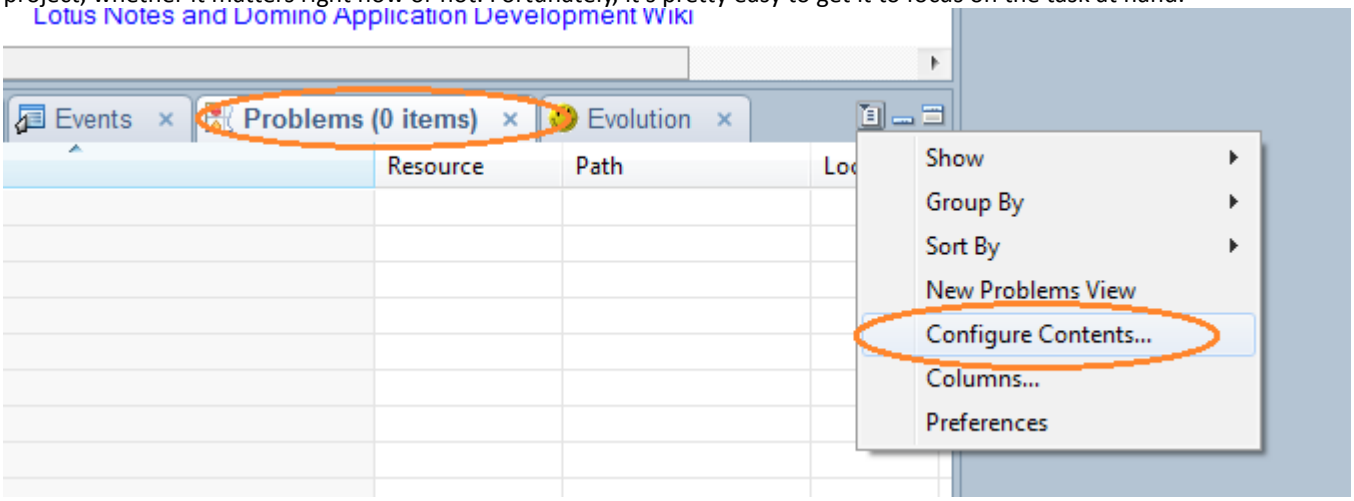
Next is "Build Automatically." Build Automatically is for people that don't understand that source code has to be compiled before it runs. So it's for spreadsheet developers, not application developers. For application developers, it's the worst kind of plague that traps noobies into thinking that everything just magically works -- a belief that last right up until they work with another developer for the very first time, and the complexities of team development hit them in the face like a cinder block. If you're reading this blog, you aren't that person, and therefore you should just turn this off immediately.



(Build Automatically is so bad that if you install our Transformer add-on for Designer, whenever you run a transform on an application, it turns off the setting immediately. And it doesn't turn it back on. Because it's a really, Really, REALLY bad idea.)

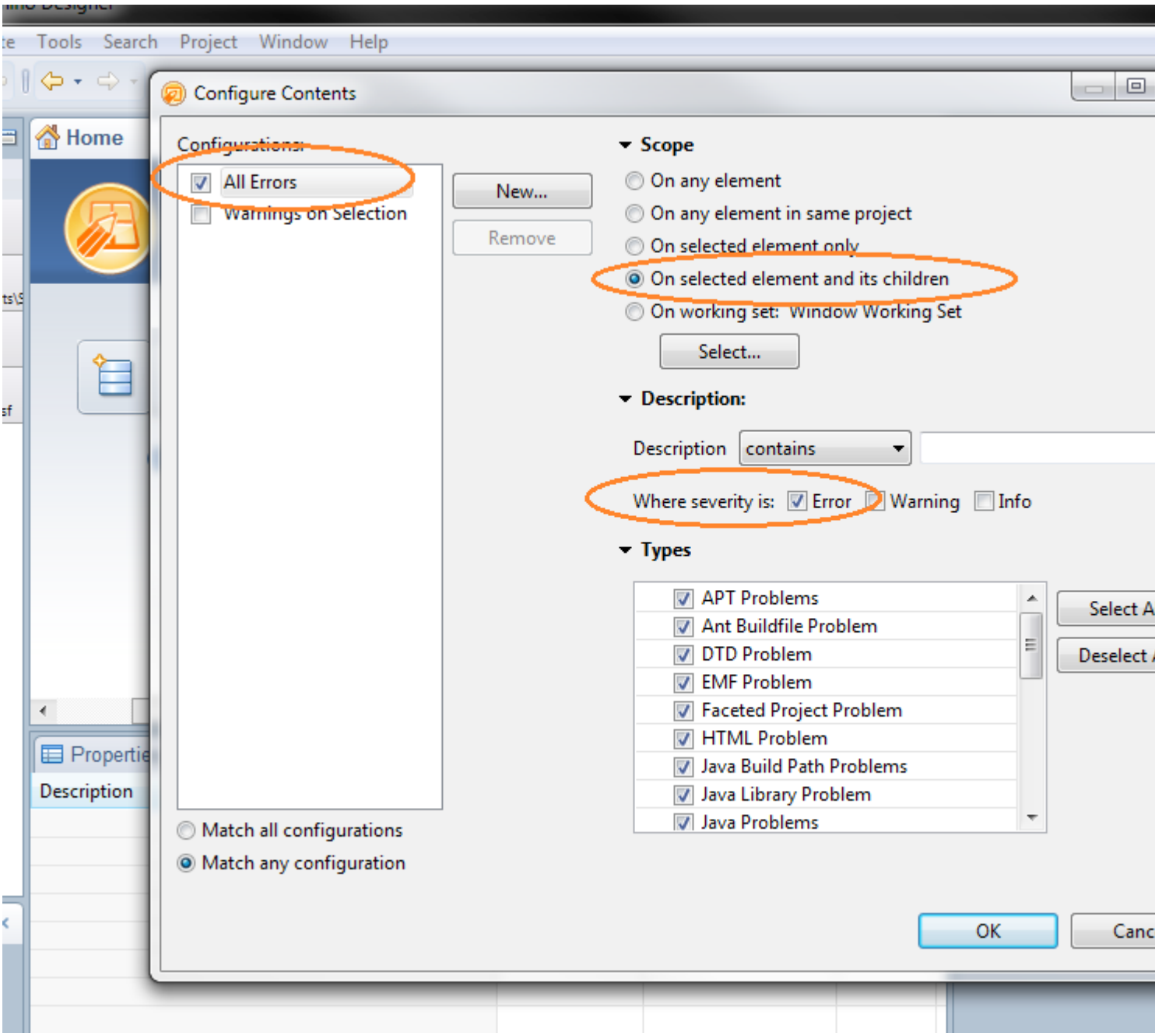
This is a good time to note that, whether you build automatically or not, it's a very bad idea to work directly against server versions of your applications in Designer 8.5. Make local templates, and either deploy them via replication, or using the standard "refresh design" command in Notes. Designer does too much automated background access against a project's file system for this to work well over NRPC. The only exception is if the server is local to the same OS that Designer is on. In that case, it's slower, but not so much that it's hard to work with.

Next, let's look at the Problems view from Eclipse. By default, this displays every issue that the IDE knows about for every project, whether it matters right now or not. Fortunately, it's pretty easy to get it to focus on the task at hand.



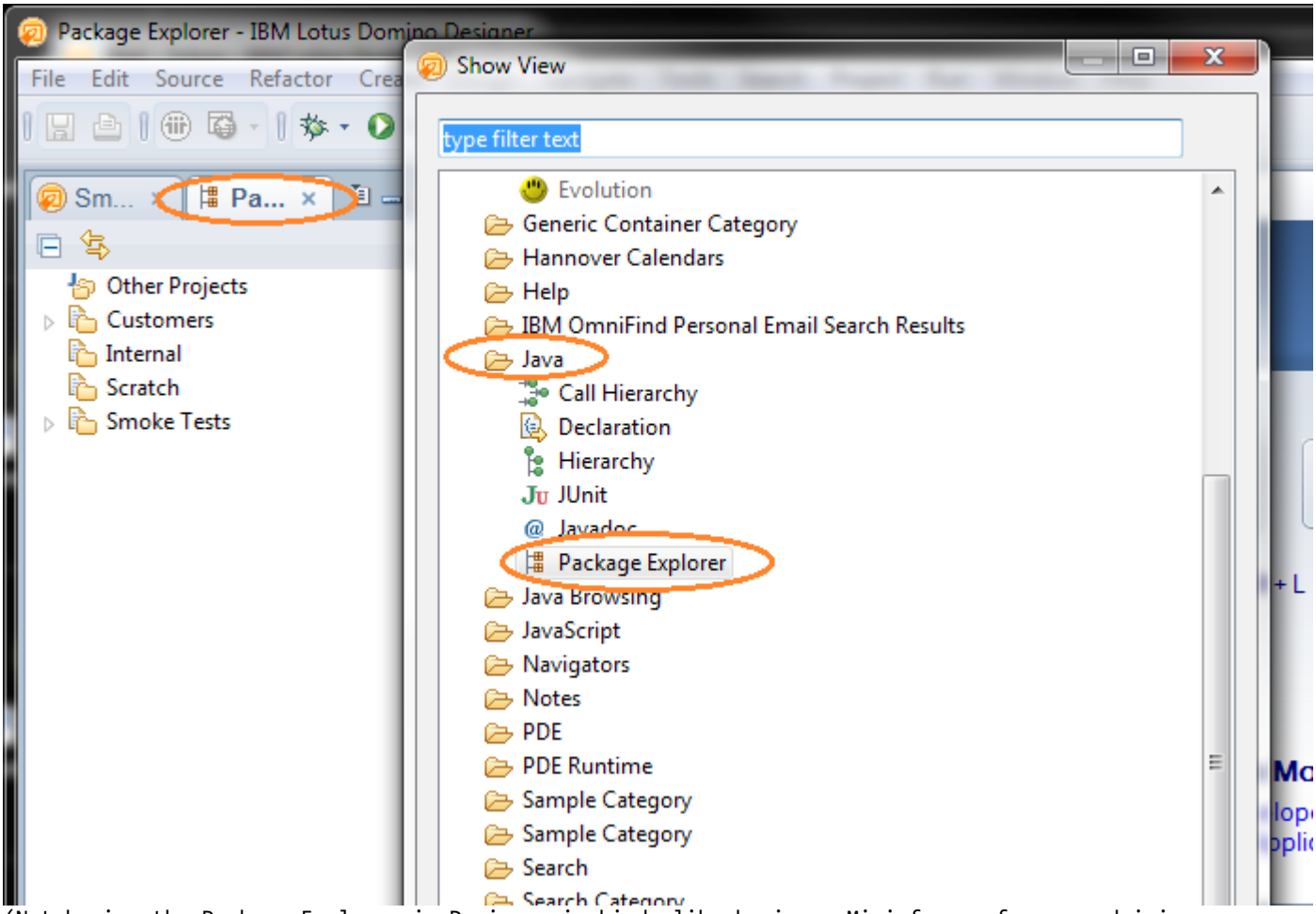
(This particular menu widget frequently reveals interesting controls. If you see something in Designer that you'd like to change, look for this control. If it's there, odds are you can get it to work like you want.)

Then change the settings to look like this:



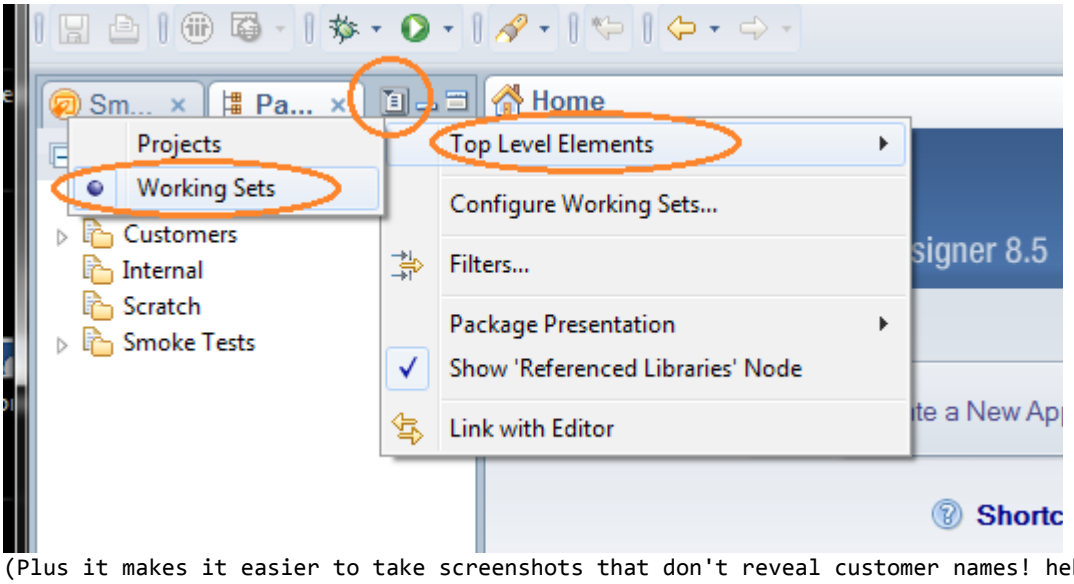
(As you can see, you can set a bunch of alternative groupings of controls. Try lots of different settings -- maybe you can improve on my recommendation!)

If you haven't already added Package Explorer to your perspective, here's how to find it. Once it's added, I usually like to drag it over to be a sibling tab to the Application Navigator itself. This is essential for serious XPages development. And once it's added, you should go to Window - Save Perspective As... and set it to Domino Designer. Yes, override the base definition of the platform, so you can make sure its still there every time you restart.



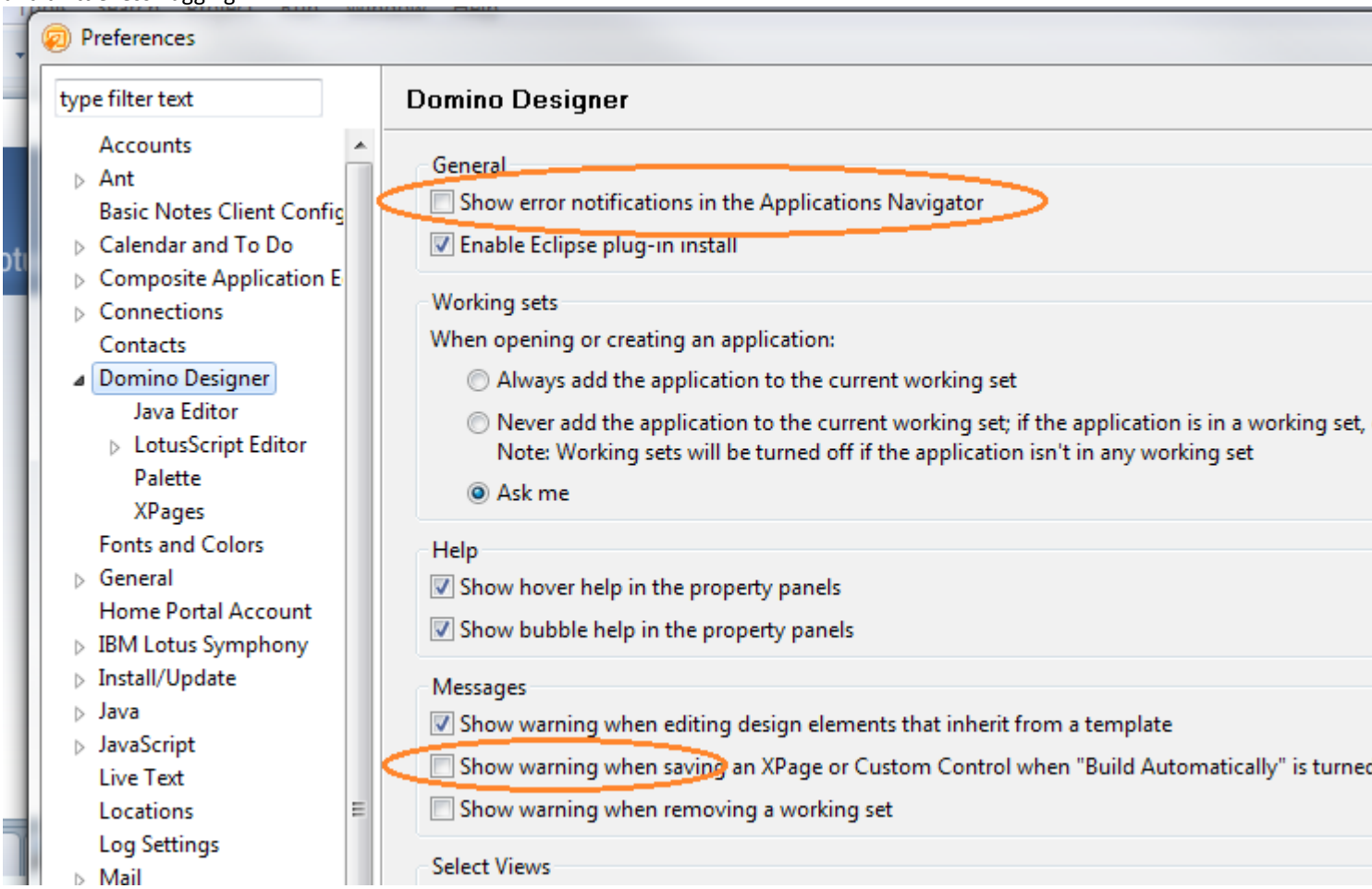
(Not having the Package Explorer in Designer is kinda like buying a Mini for performance driving, but getting an automatic transmission. If you're gonna drive a performance car, DRIVE the thing!)

Once you've added the Package Explorer, be sure to set it to display your Working Sets so they have double-duty. Also, once you do this, if you do a right-click - Build on your Domino projects, they will happen with a proper UI that details what's going on instead of the blackhole of the App Nav build process. Once you make the change, you'll never want to go back.



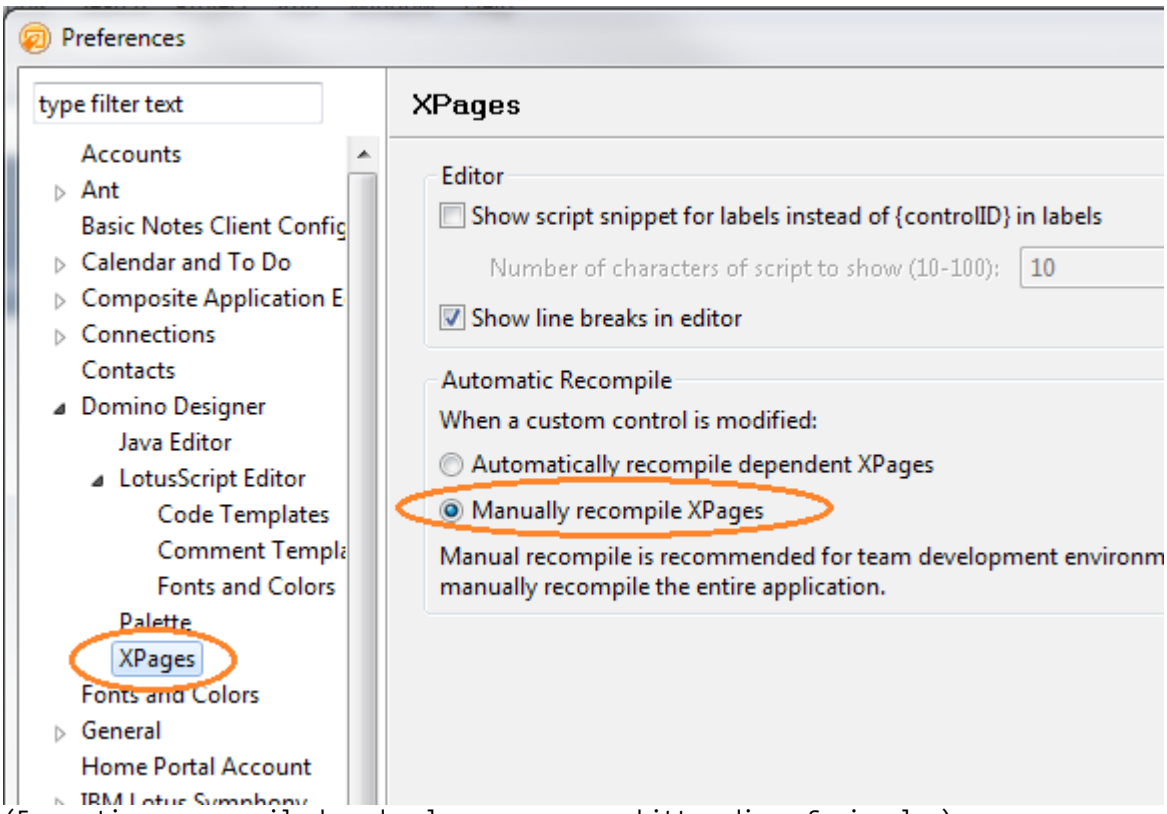
(Plus it makes it easier to take screenshots that don't reveal customer names! hehe)

Designer itself has a few preferences worth noting. I like to set the following just to have the IDE doing a little less work and a little less nagging.



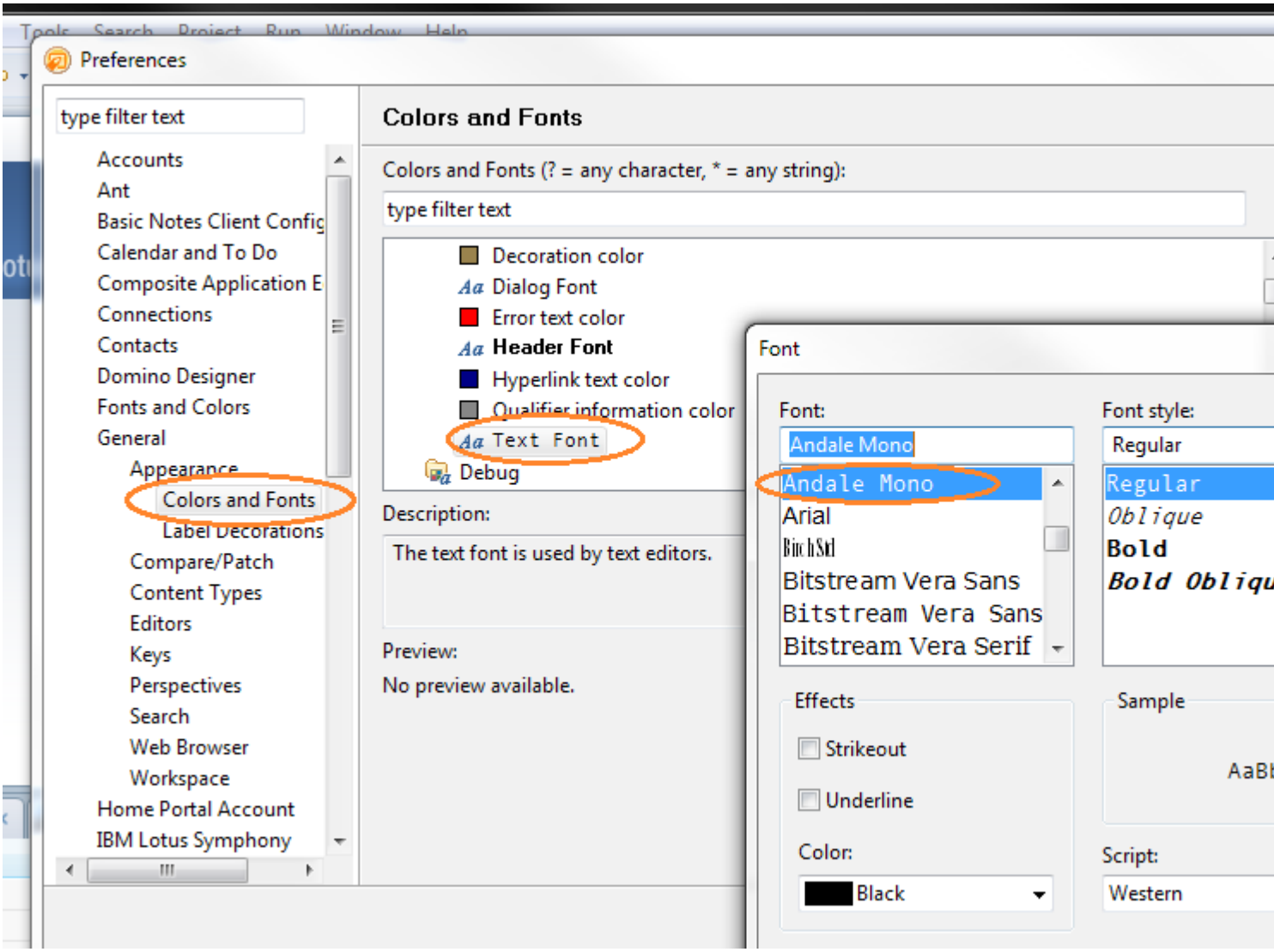
(Maureen says Designer is female. These settings will make it a little more masculine.)

Also make sure you turn off more automatic compilation, because you're a real developer, right?



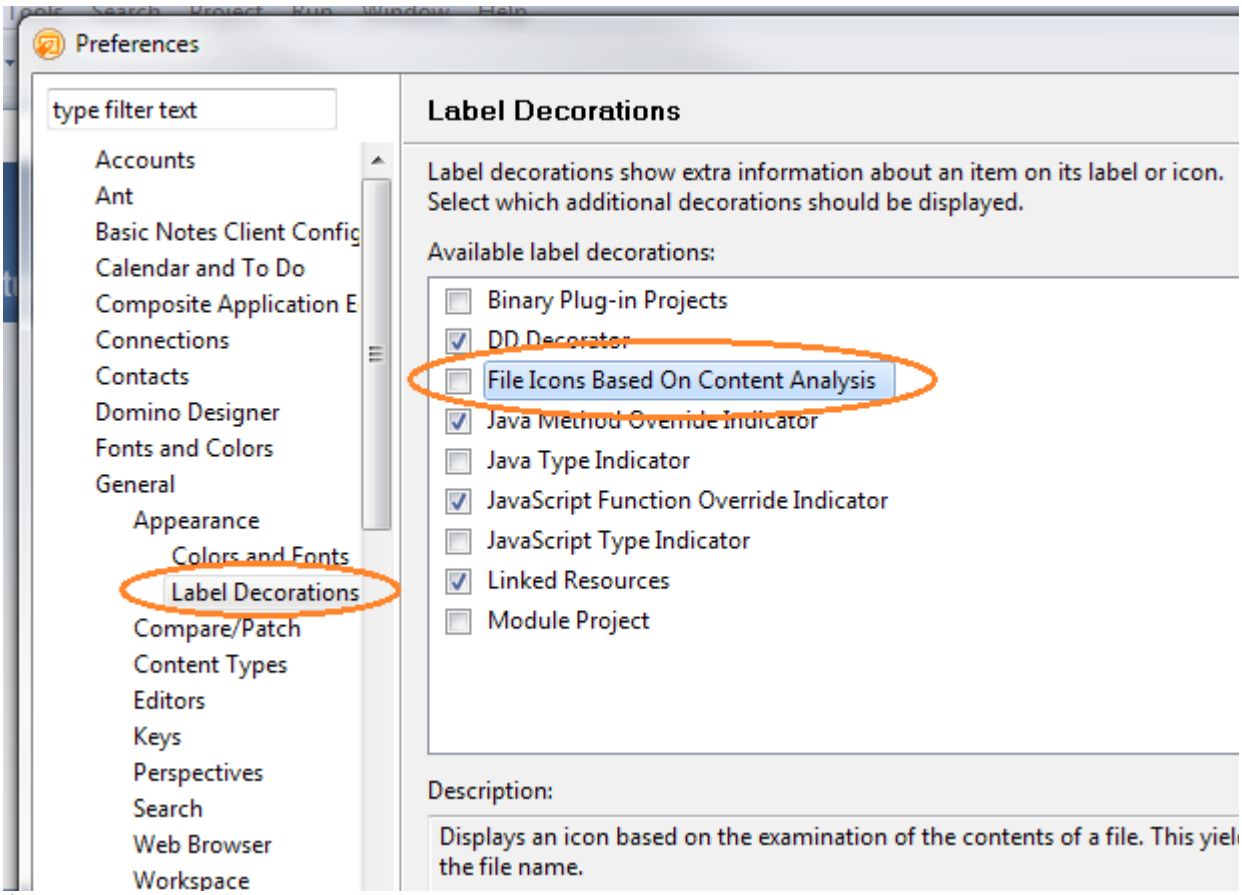
(Every time you compile based only on a save, a kitten dies. Seriously.)

By its default settings, Eclipse displays normal text editing in Courier. I don't really have anything against Courier except that IT SUCKS. So if you really want to work source code in Designer, set your default font to something readable, like **Andale Mono**.



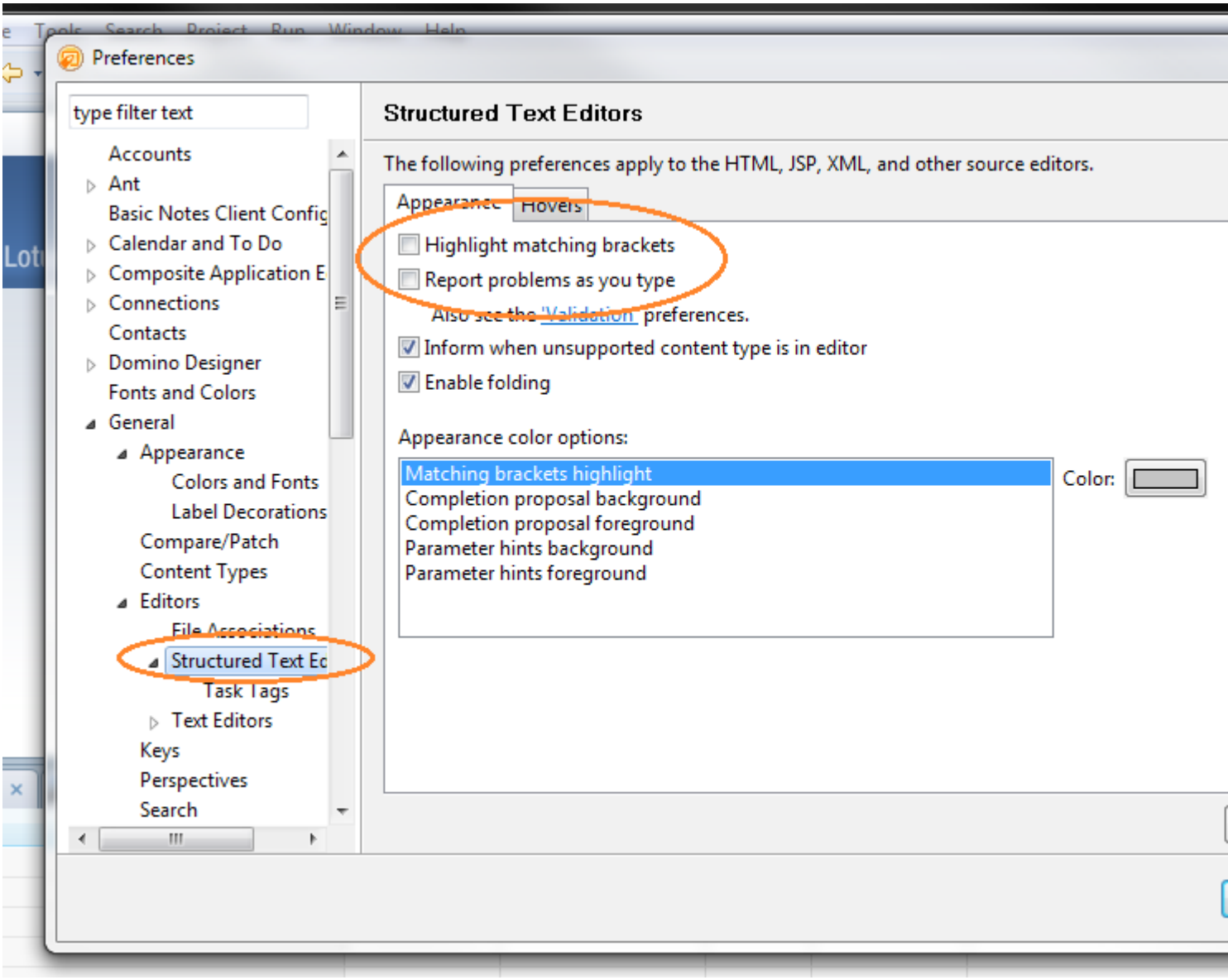
(If you don't believe me, Google Courier and Andale Mono. It's a documented fact that Andale Mono is a better typeface for source code.)

Sometimes surprising things can slow down your experience in an IDE. One of those is when a tree navigator thinks it needs to show you the ideal representation of every imaginable piece of content. This is pretty much totally unnecessary on Domino projects, so there's no reason to go through all the extra file I/O it requires. Turn it off.



(If you don't turn this off, Eclipse is actually opening every file it displays to make sure that the content type matches the extension. Can you imagine the workload involved?)

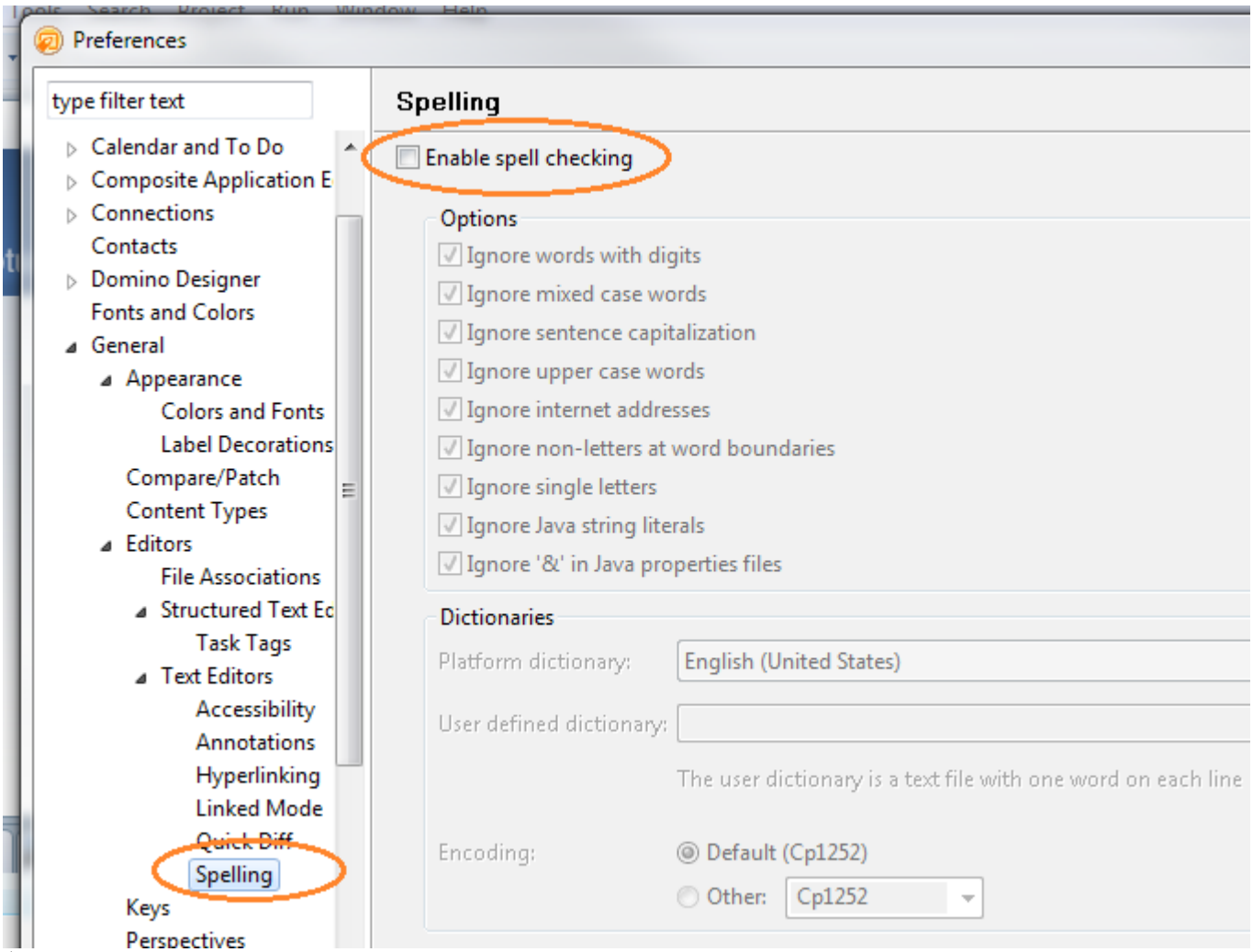
If you're an XPages developer who works a lot in the Source tab, you've probably been frustrated by slow downs in the Editor. Well, it turns out that some of these things are just default behaviors of the editor itself, and you can control them.



(Leave these setting on if you frequently have trouble keep track of your syntax nesting in XML.)

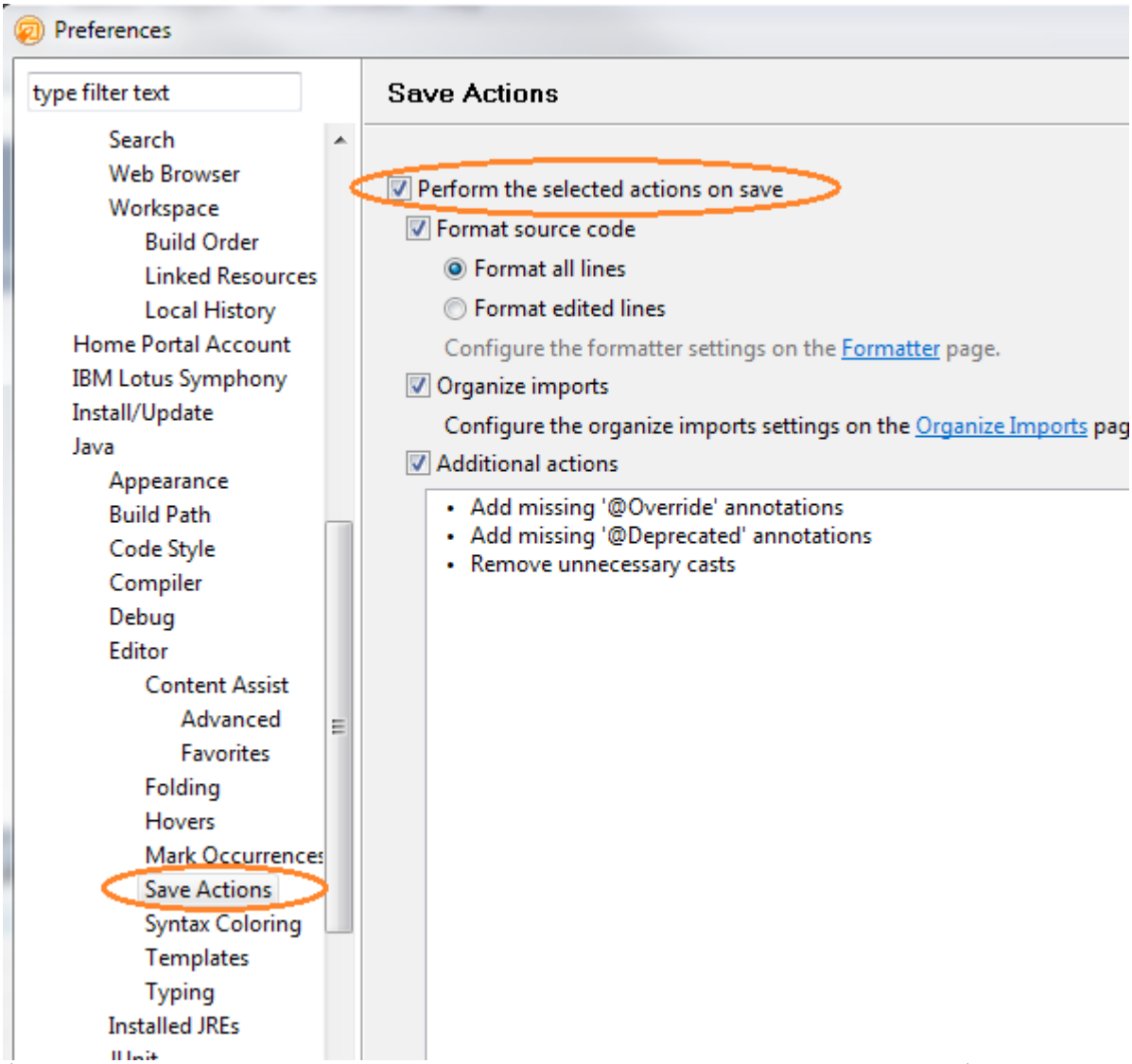
Note: the single best thing you can do to make sure you don't have a bad experience in the XPages source pane is to NEVER write href references to files with spaces in the name. This causes the editor to try to resolve the reference itself, and if the file doesn't exist because you haven't typed the full name, it will spend a lot of time trying to resolve it. You'll perceive this as "suddenly the keyboard doesn't respond, and letters are only added every two seconds." Similarly, if you want to comment out some content, highlight the whole block and hit Ctrl + Shift + /.

Spell-checking your source code is almost definitely useless, so turn it off.



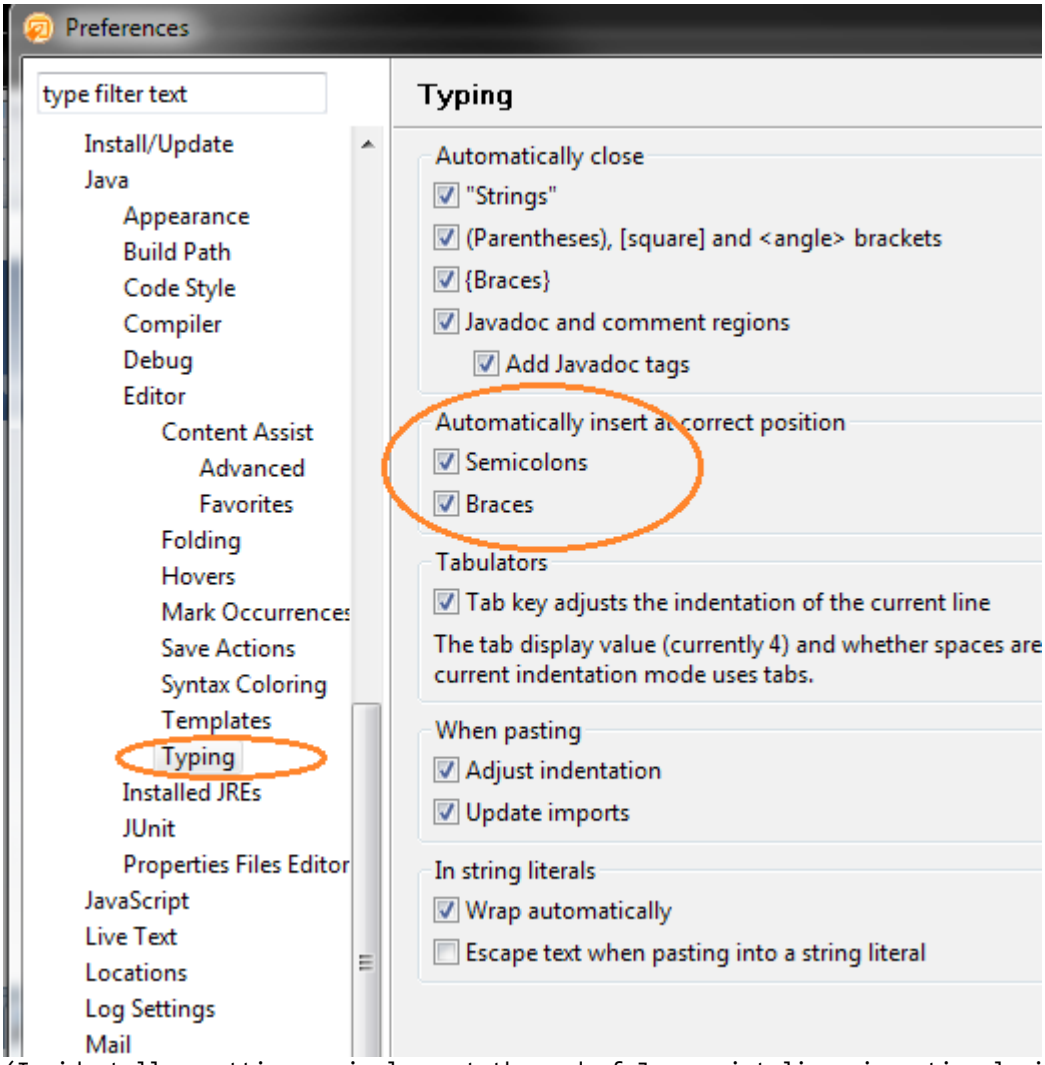
(Okay, it's a good idea if you've added all your variable names to your local dictionary. And if you did that, you are a pedantic idiot.)

If you write Java code directly into your XPages projects, then some simple Java behavior changes make sense. I, for one, am fanatical about source code formatting, so I like to make sure this happens at save.



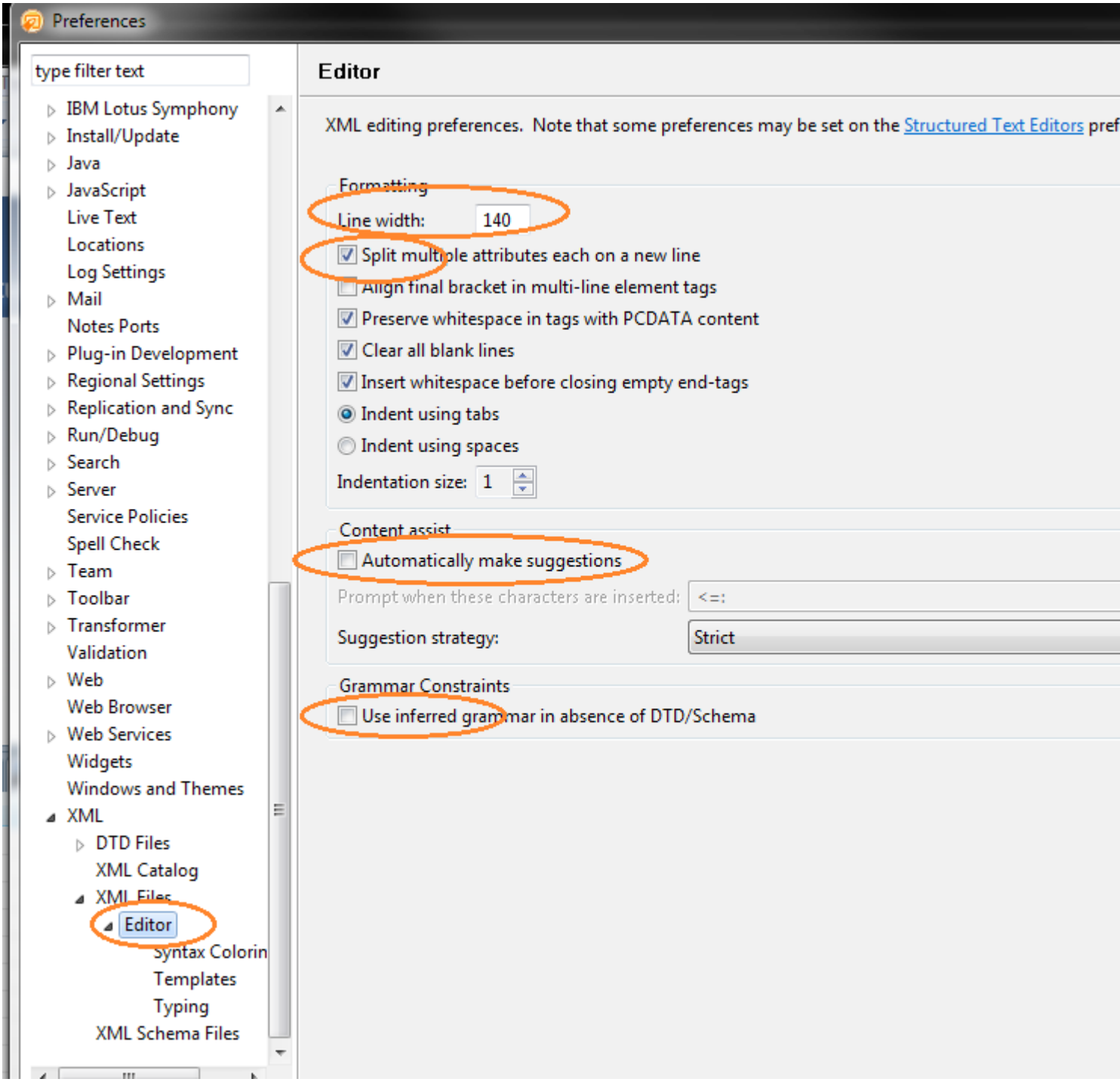
(There's a crazy number of possible controls here. You should check them out.)

I also like to let the editor do a little extra work for me.



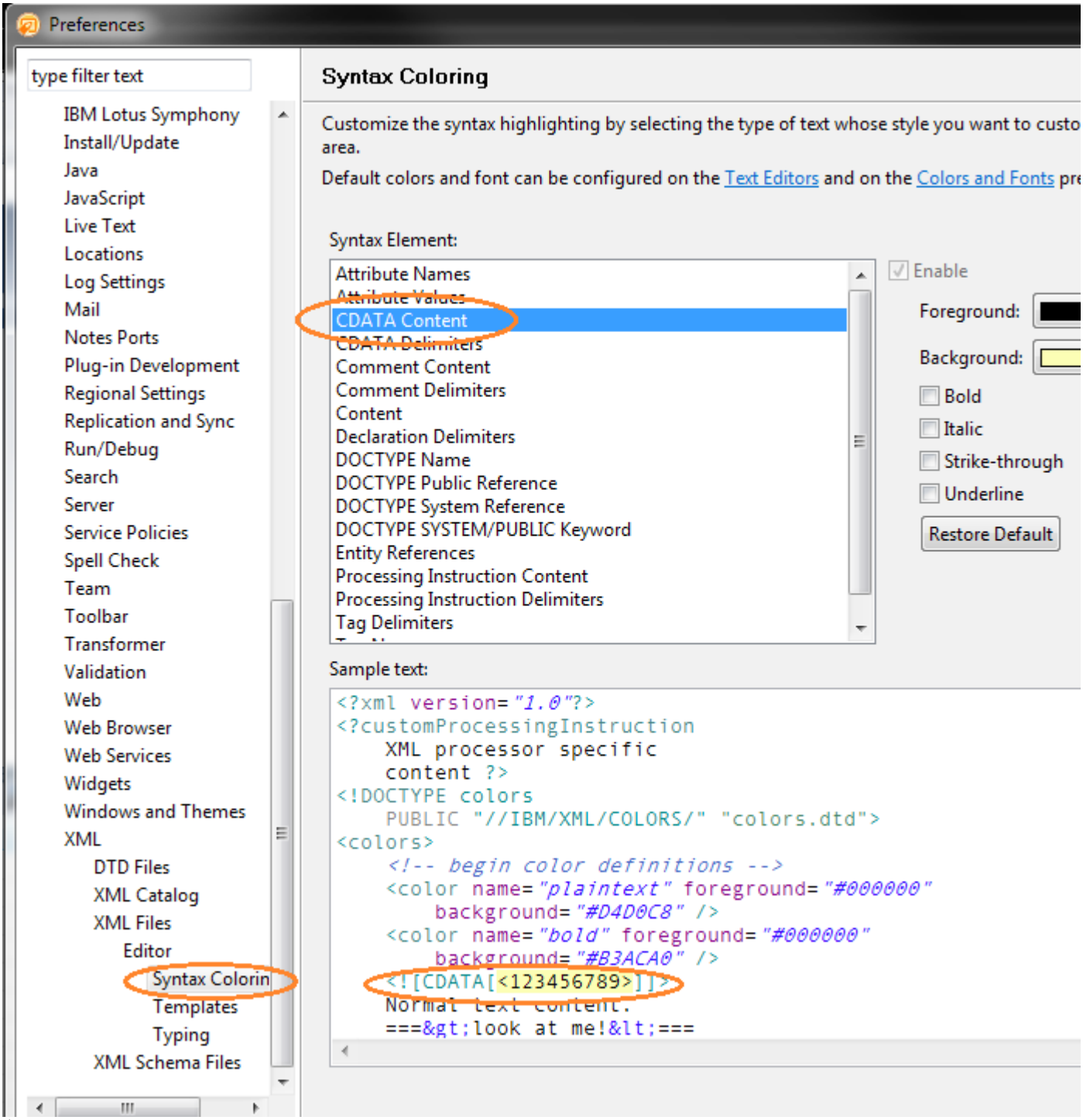
(Incidentally, putting semicolons at the end of Javascript lines is optional, if you're an amateur. If you don't wear diapers to work, then you add the delimiter to the end of a statement.)

Many of the most important controls are in the XML editor, because these affect the Source tab in Xpages. Mostly, you want to make that Editor better for widescreen monitors, and stop trying to tell you what to do.



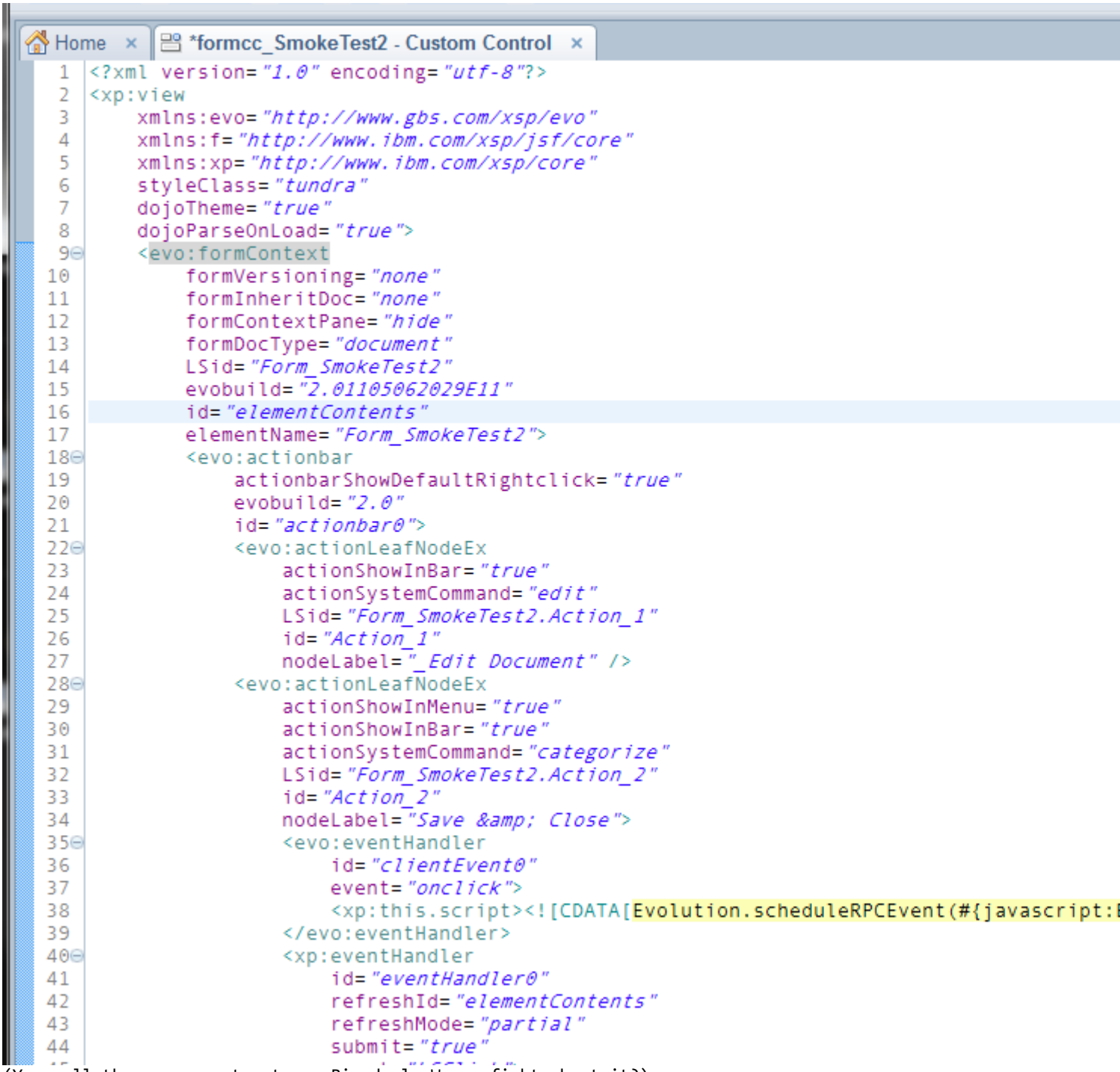
(Jeremy Hodge recommended the 'split multiple attributes' setting, and I thought he was crazy. Until I did it. Now I couldn't imagine working any other way.)

If you want to make large blocks of Javascript easier to process in the Source tab, then you need to make sure they're visible. I've found the following small change to work out very well.



(Okay, to be honest, the highlighted code is all Javascript, and so I throw up in my mouth a little. But it's also better for revealing bad behavior, so it's worth the occasional reflux to spot JS code easily.)

Once you've tweaked all these settings, you should find XPages content much easier to digest visually...



(Yes, all these are custom tags. Big deal. Wanna fight about it?)

Also, if you haven't installed the **Source Code Enablement plugin for Designer**, do so RIGHT NOW. (Update: this is no longer needed with Designer 9.0) And if you don't have a source code management server available, then build one immediately. You can use SVN or Mercurial and get support from Eclipse.org, or you can use git, and **get support from OpenNTF.org**. Or you can use Github or Bitbucket. SCM is absolutely mandatory in multi-developer projects, and if you aren't using it, you're taking needless risks with your applications.

That's it for this particular article. There are certainly other things you can do to make Designer do even more, whether with **CodePro** or **Aptana** or **Mylyn** or **JUnit**, but these are beyond the scope of just tweaking what Designer does today.