Troubleshooting An Embedded Sametime Install

by Julian Robichaux, panagenda
originally published on socialbizug.org, November 2013

I was testing the new IBM Sametime 9 client on a few different virtual machines recently, and there was one IBM Notes client I had where the embedded Sametime client would not install. I finally got it working (yay!), but it wasn't obvious how to troubleshoot and fix the problem so I thought I would share some of the steps I went through and describe how the whole process works.

While I am specifically talking about the Sametime 9 embedded client install below, the information is also relevant to any kind of Notes client plugin that uses an MSI or an EXE to install, like the IBM Connections plugins.

Step 1: An Install That Hangs
To install the Sametime embedded client, I extracted the sametime.embedded.addon.win32_20130911-1333.zip file from the whopping 2+ GB SametimeConnectClient download from the IBM software catalog. After unzipping this file, I closed the Notes client and ran the SametimeEmbedded.msi installer. I went through a few verification screens and finally got to a progress bar dialog that seemed to indicate that the embedded client was indeed installing.

But then it got to this screen and froze up:
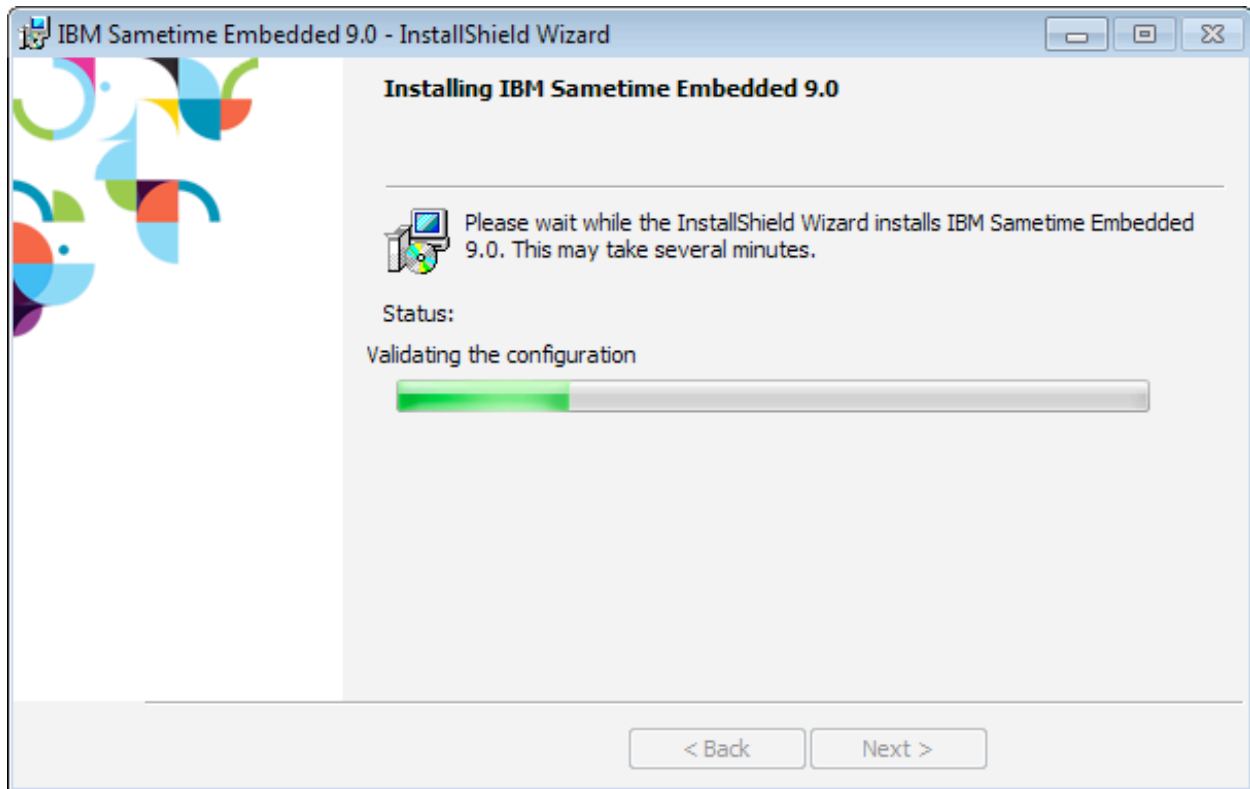
After 5 or 10 minutes of staring at the "Validating the configuration" message and watching my CPU stay pegged at 90%, I decided that the install was hung and I killed it in Task Manager. Killing the install didn't stop the high CPU utilization though, and I also had to kill the notes2.exe process:
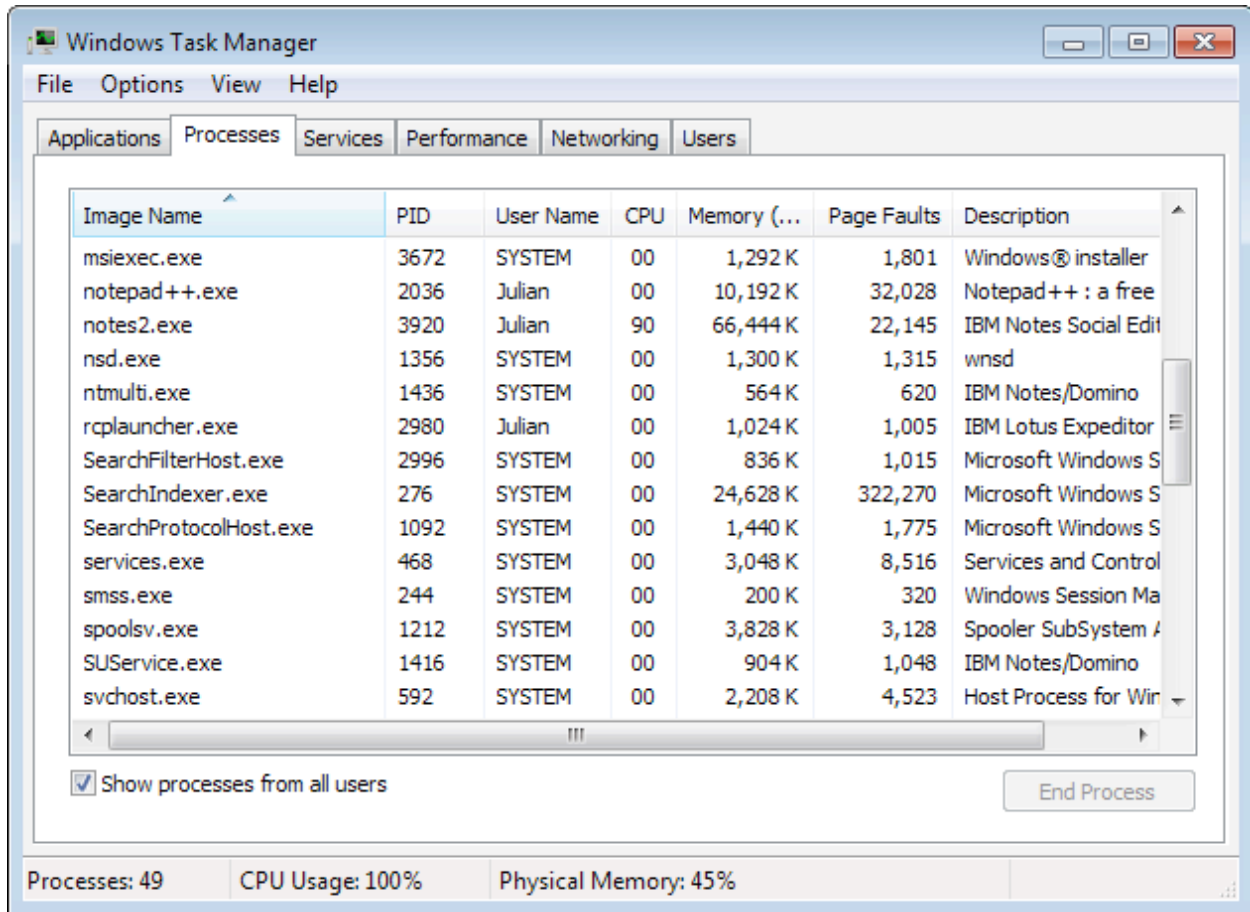
The setup I had was an IBM Notes 9.0.0 client on Windows 7. I have used this VM for testing many times before, so I knew that the client itself worked and I had been able to install plugins and use Sametime in the past. I was also relatively sure that it wasn't a permissions issue because the Windows security dialog popped up and asked me to continue. So where to look for problems?

Step 2: Finding the Logs
For this type of MSI/EXE based plugin install, it is helpful to understand how it works before you start troubleshooting.

The Expeditor (Eclipse) part of the Notes client uses a process called Provisioning in order to install plugins. An MSI/EXE installer works by running the Expeditor launcher in the background, using special command line options that trigger the provisioning operations. When the provisioning is finished (either successfully or unsuccessfully), the Notes process associated with the Expeditor launcher is shut down.

This is why you can't have the Notes client running when you use the installer, because the running Notes client would conflict with the background Notes client that is launched

and used for provisioning. It also explains why we had to kill the notes2.exe process before. More importantly, it is an indication of where to look for log files about what happened: in the [notes_data]\workspace\logs directory, the same place where the normal Notes Eclipse logs are found.

I sorted the files in my logs directory by date to show you which ones changed during the failed install process:
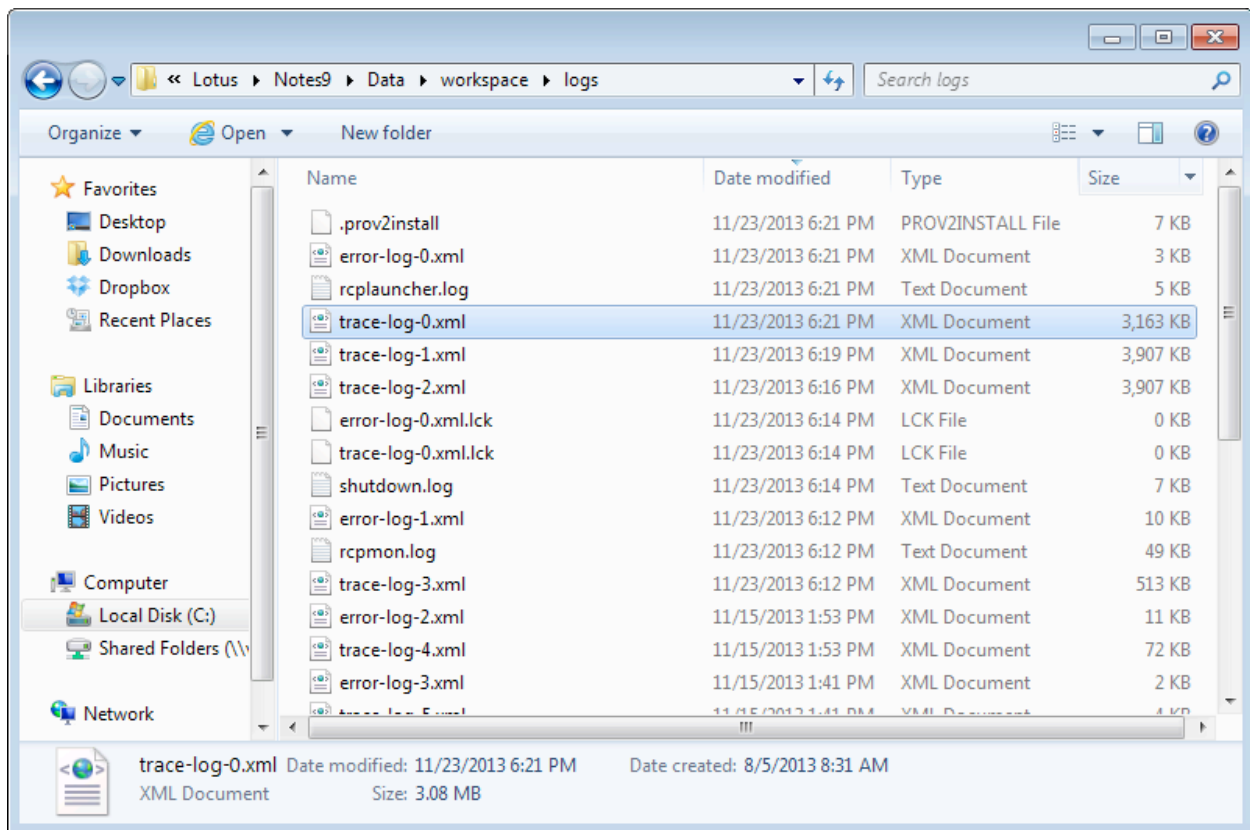
The .prov2Install and rcplauncher.log files give a tiny bit of information about what happened during the install process, but nothing really useful. The good stuff is in the trace-log files.

Step 3: Looking at the Log Files
If you open the trace log files as XML files, either in a text editor or an XML editor, they are pretty hard to read. The easiest way to look at them is by opening them in Internet Explorer directly from the logs directory. YES, I SAID INTERNET EXPLORER.

This is because IE will use the XSL stylesheet that also resides in the logs directory to format the XML log files for easy reading. Old versions of IE would do this automatically, but newer versions (starting with IE 8 or 9) have the local security settings cranked up a bit higher so we have to do a little trick.

I had IE 10 on this particular machine. When I initially opened the trace log file in that browser, it looked like a bunch of unformatted Times New Roman text. To trick IE into using the XSL transform, we need to put it into compatibility mode. Here's how:
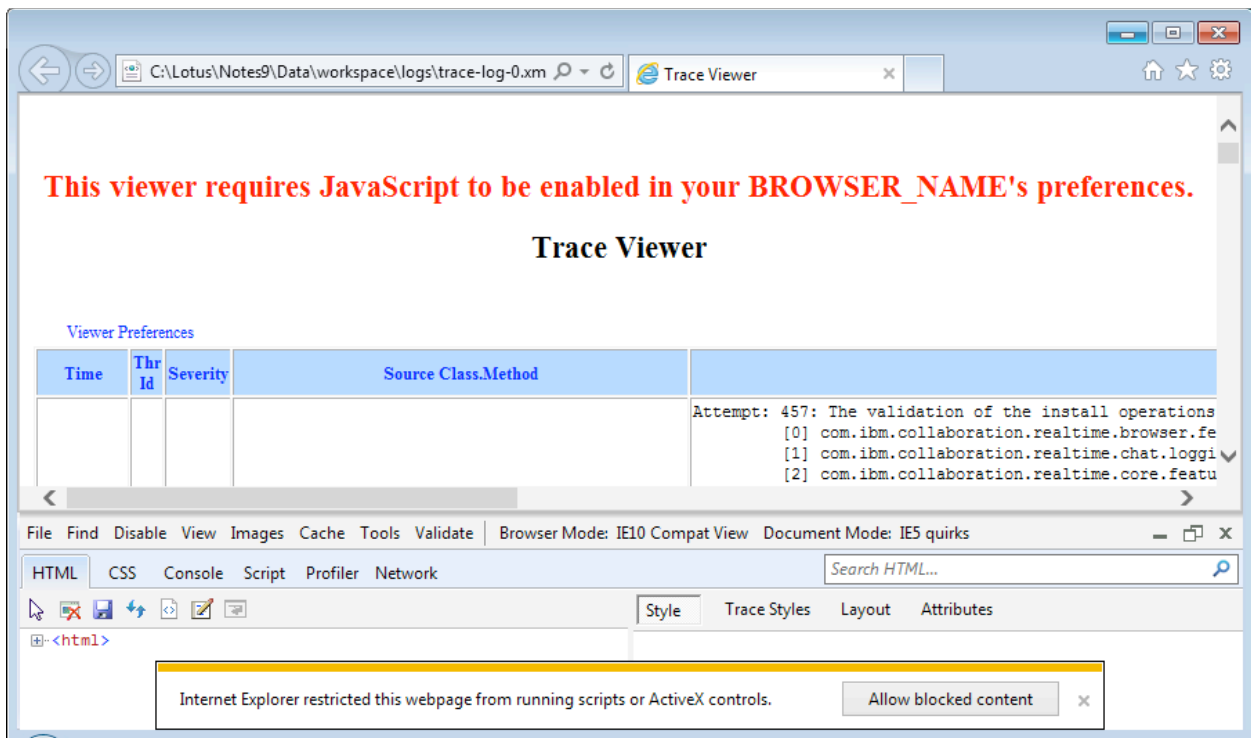
1. Open the Developer Tools from the Tools menu.
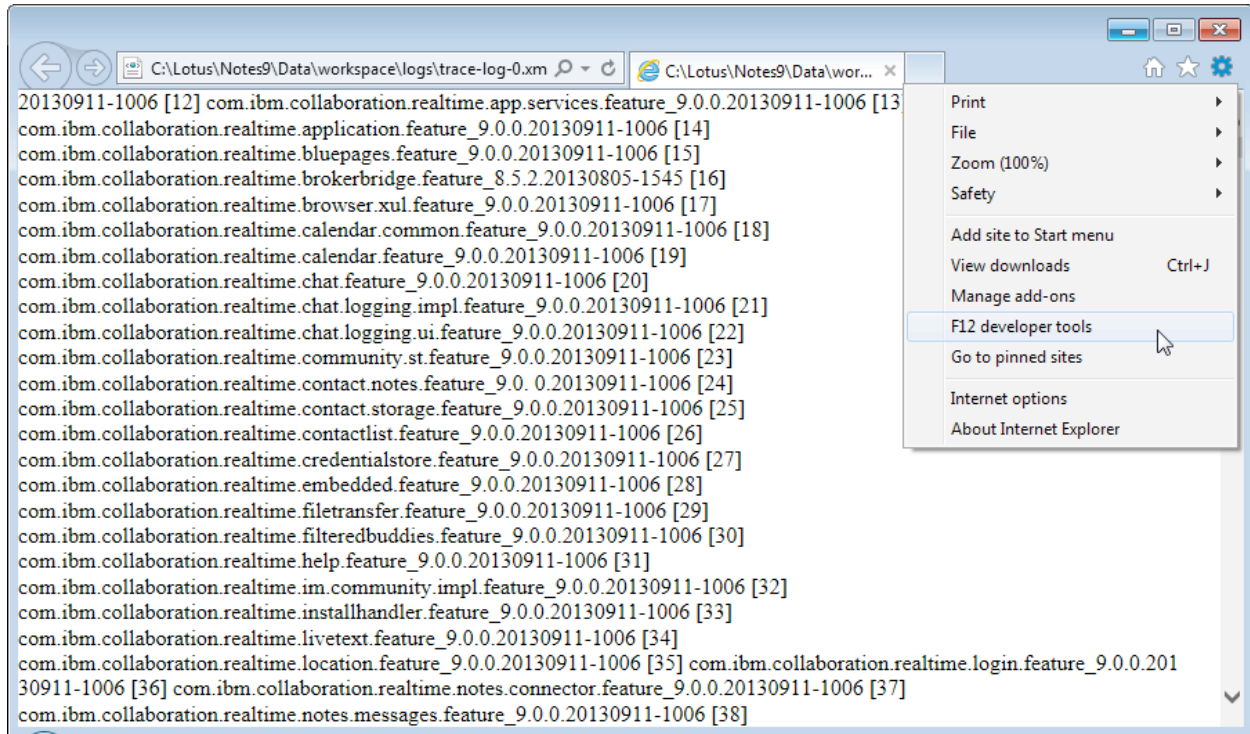
2. Switch the browser mode from "Internet Explorer 10" to "Internet Explorer 10 Compatibility View".
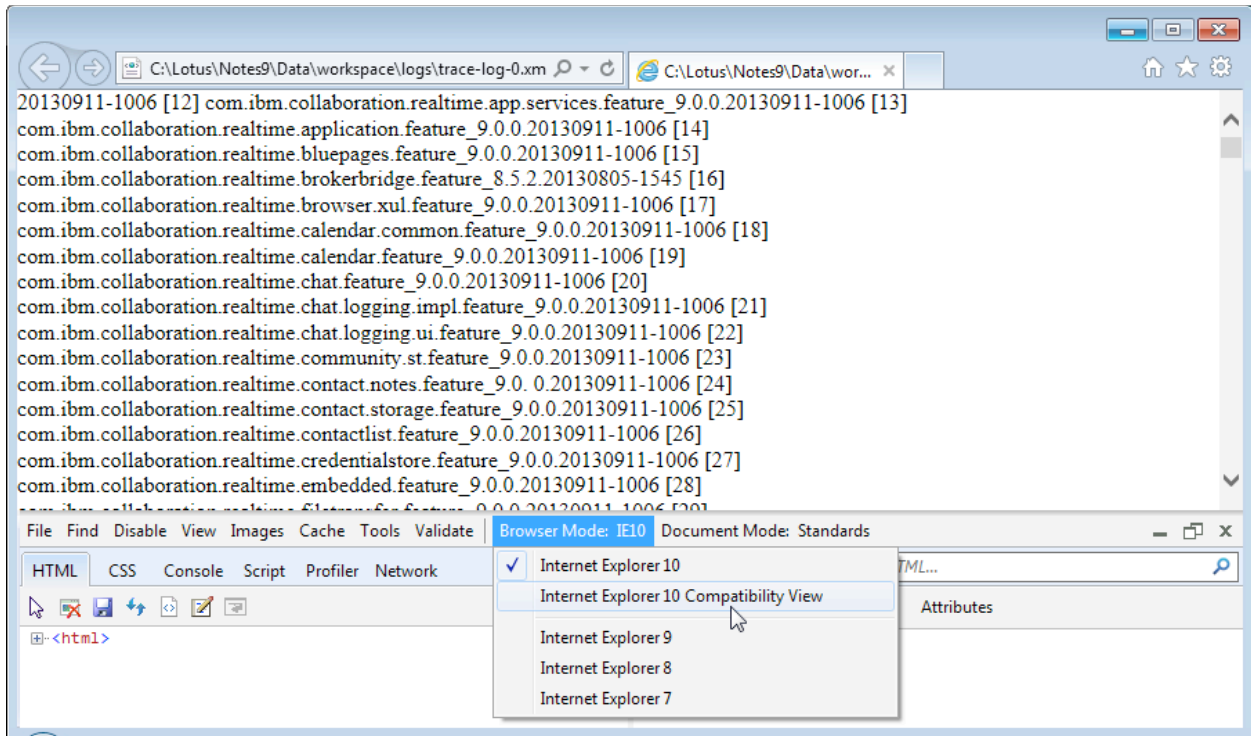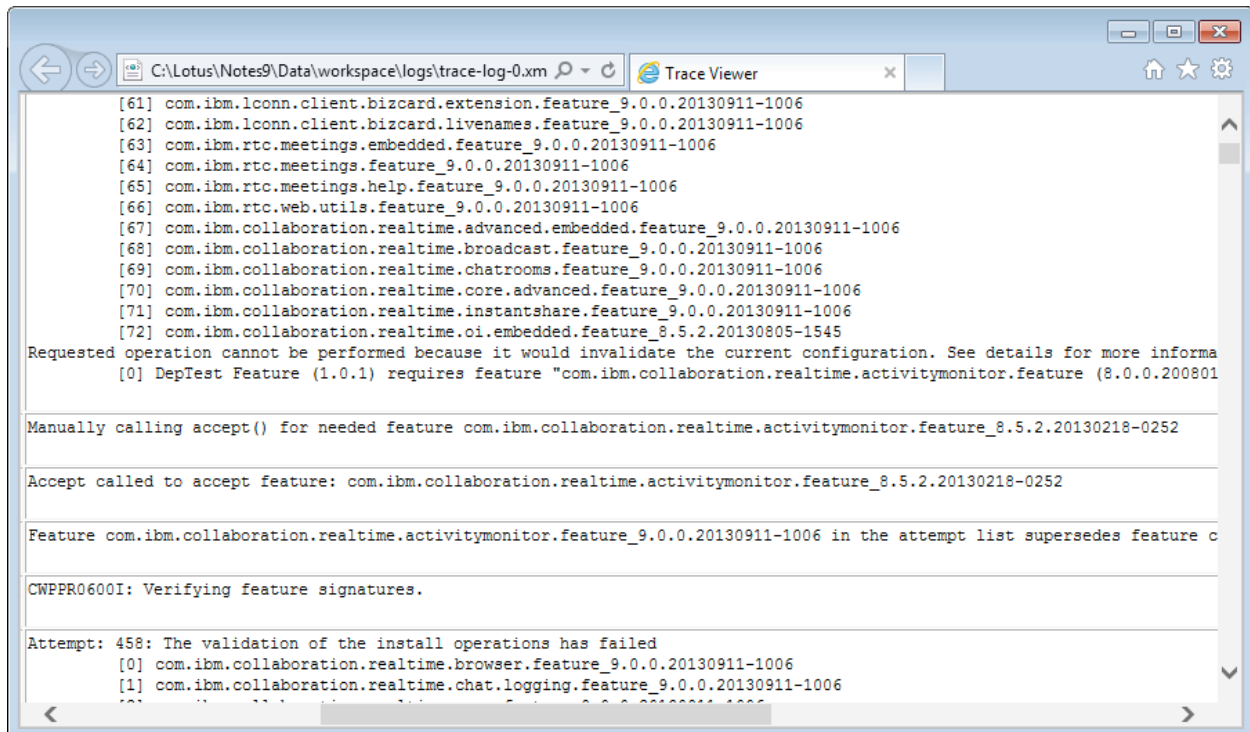
3. Watch in amazement as the plain text in the browser magically transforms into a nicely structured table with alternating color rows.

Step 4: Understanding the Logs

The trace log really ends up showing the same 5 messages over and over. This is because the installer keeps attempting the install, then the install fails, and then it attempts the install again, and it fails again, and on and on. The messages even have an incrementing number next to each install attempt so you can keep count. In my case, the installer was on its 700th attempt before I finally killed it.

In a pleasant turn of events, if you actually read the install message it tells you exactly what caused the install to fail:

Julian's Secret Troubleshooting Tip: reading the error message often tells you what the error is.

Here we can see the message "Requested operation cannot be performed because it would invalidate the current configuration" followed by:

[0] DepTest Feature (1.0.1) requires feature "com.ibm.collaboration.realtime.activitymonitor.feature (8.0.0.20080130-1459)", or compatible.

OOPS! Do you remember that I told you I use this VM for a lot of testing? Well, one time long ago I was testing the effect of dependency declarations in plugins, and I created a plugin (cleverly named "DepTest") that had several dependencies on other plugins. One of the dependencies I used for testing just happened to be a Sametime plugin, and

because of the way I declared the dependency the provisioning engine wouldn't allow the Sametime installer to break my plugin by replacing the 8.0 version of the activitymonitor feature with the 9.0 version.

Unfortunately, this also put the installer into a death loop of trying to install over and over again. This seems to be a programming bug to my untrained eye, but it's at least a bug I can work around. Just uninstall the troublesome DepTest plugin (which I don't really need anyway) and try again.
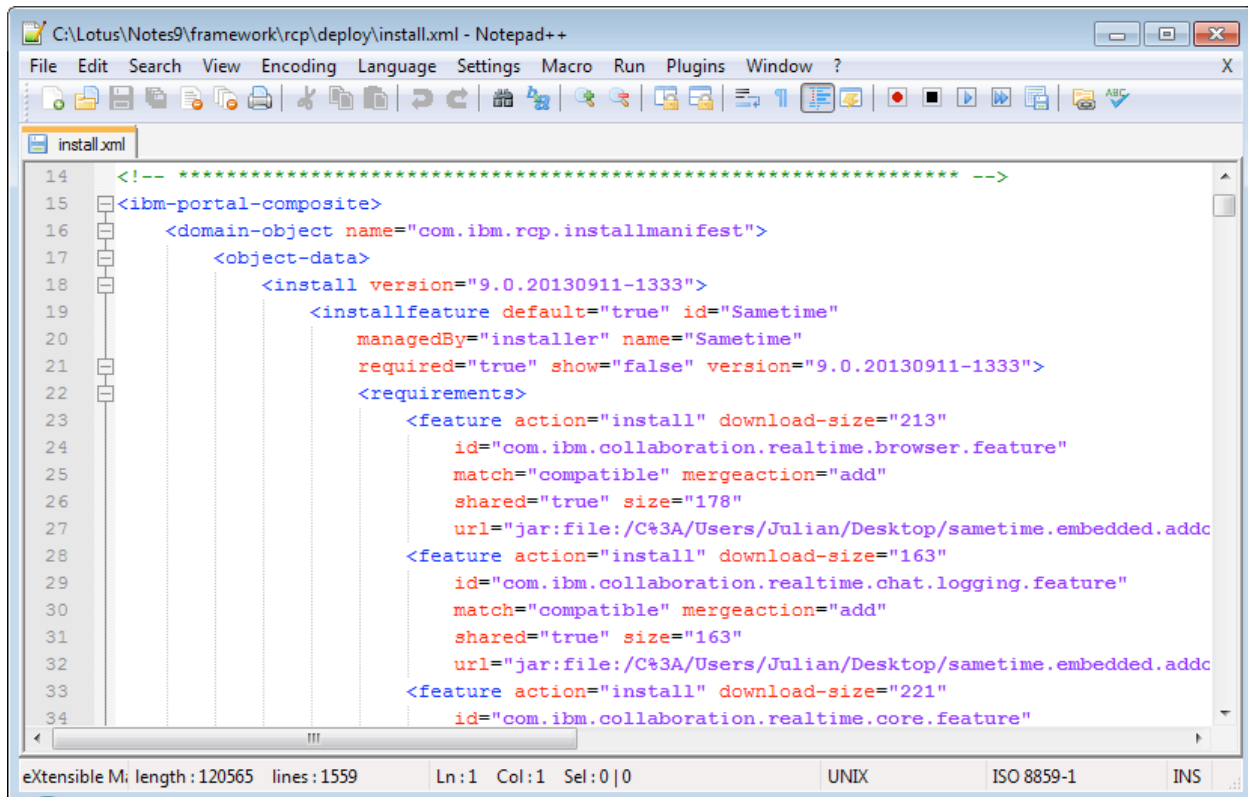
Step 5: Why Won't My Notes Client Start?
In order to fix this I just have to open my Notes client and uninstall that plugin and… wait, why won't my Notes client start? Now when I launch Notes, I login and it just hangs at the splash screen with the "Loading… please wait" message. Hilariously, when the "Loading" progress bar gets all the way to 100% it starts over. I love when progress bars do that.

To understand what's going on here, you need to navigate to the Notes\framework\rcp\deploy directory (and you'll also want to kill the still-running Notes processes in Task Manager).
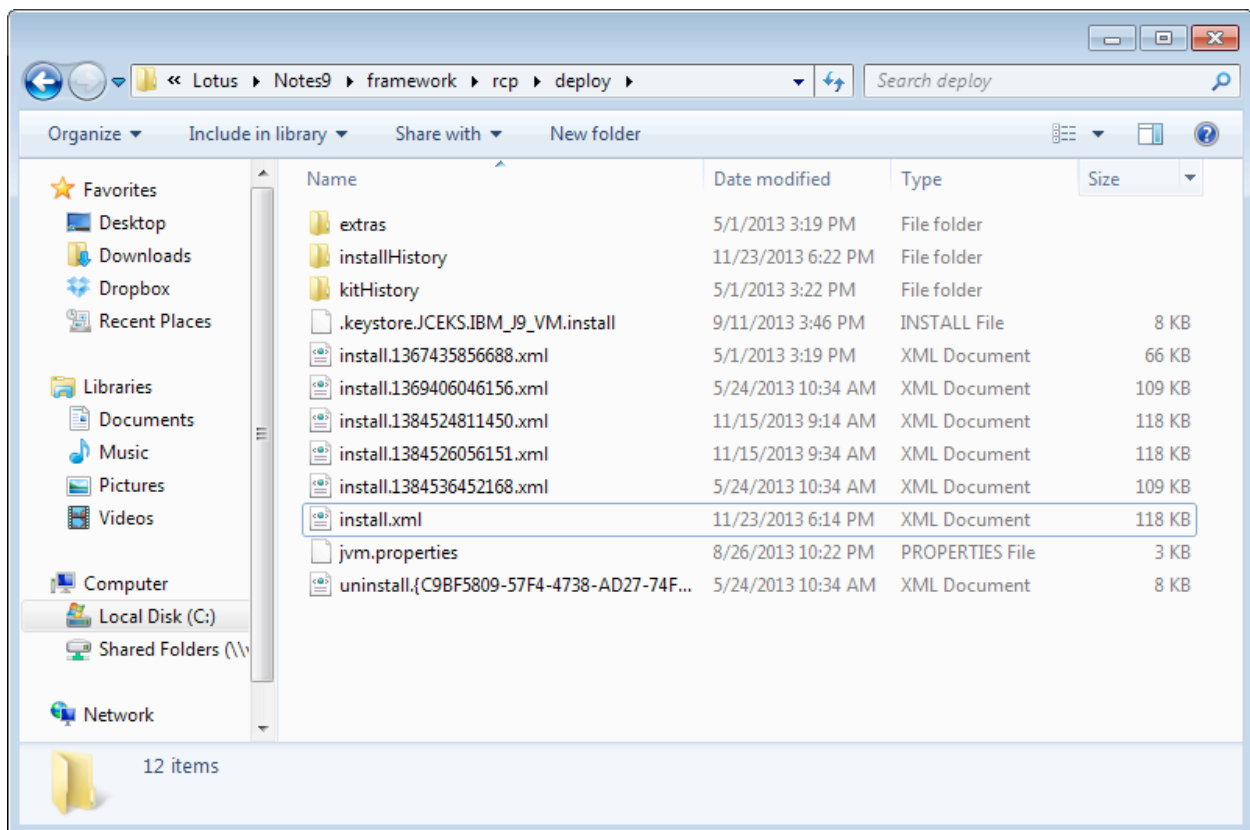
As part of the provisioning process, the new Sametime embedded client plugins were added to the install.xml file. When the Notes client starts up it reads the install.xml file and tries to use the new Sametime plugins and… well, at this point I'm honestly not sure what happens. I don't know if it tries to reinstall the plugins (because it still knows the path to the installer files) or if it just gets sad and curls up into a little ball or what. All I know is that Notes won't start.

However, the Notes client keeps backups of old install.xml files. A lucky break! All you have to do is find an old install.123456789.xml file in the deploy directory that does not contain references to the new Sametime plugins and rename it to "install.xml". Now the Notes client should start and all is back to normal.

In Conclusion
While I seriously doubt that anyone else would have exactly the same problem as me in this situation — I am quite sure that no one else has my DepTest plugin — it's always good to know how things work, how the pieces fit together, and especially how to troubleshoot. And even if no one else does have my troublesome plugin, it's not too much of a stretch to think that there might be other plugins in the wild that have Sametime dependencies and might exhibit similar behaviors.