OISV Home

Logged in as: Search Engine
Profile | Logout

- Articles
- Forums
- About Us
- Contact Us

GoDaddy Gotchas

In early 2006, I began work on my first commercial web application. My plans were simple. Develop a Java-based application using industry standard practices and open source components. The application would begin life on a GoDaddy shared hosting server. Once traffic picked up, the application would step from shared hosting to virtual hosting to dedicated hosting, and perhaps, one day, to a co-located server.

Two months later, armed with a WAR file and a credit card, I signed up at GoDaddy.com for shared hosting. That's when the fun began.

The carefully crafted Java web application, which ran so well on my development workstation and laptop, did nothing when deployed on the GoDaddy server. Really, it did nothing. Off I went to read the support pages.

Oh, you have to wait until GoDaddy bounces the Tomcat instance (every night at 1:00 a.m. Arizona time). I dutifully waited until the next morning. Eagerly, I opened my browser to the server and... Nothing. Again.

I checked the server using the FTP applet provided in the GoDaddy Control Panel. The WAR file had been unpacked. Everything looked okay. What could be the problem? Since I was using Log4J, I knew I could find the information I needed in the logs. Hmmm... No logs.

(Side note #1: A few weeks into my GoDaddy experience, I switched to FileZilla for FTP functionality, something I should have done the first day. The FTP applet is severely lacking.)

Faced with a failing web application, no log files, and no access to the server logs, I started with the basics. First, I built a 'Hello World' web application and deployed it on the server. It worked fine. Next, I added Struts and deployed. Again, no problems. Next, I added Tiles. Good. Lastly, I added Log4J. Finally, the problem reappeared.

Note that each of these builds took 24 hours to test since I had to wait for the Tomcat instance to bounce. On some days, I went to check the application only to discover that WAR file had not been unpacked. A phone call to

GoDaddy revealed that my server was scheduled to bounce at 10:00 a.m. on that particular day. No reason why, though. I checked the server later that evening and, sure enough, the WAR file had been unpacked.

(Side note #2: GoDaddy has been extremely helpful in each instance. Although it was a toll call, I get free long distance on my cell phone, so it was essentially free. Also, the Help Desk employees were polite, well spoken, and intelligent. With the exception of one instance, noted below, they were able to fully answer my questions.)

Now I knew that Log4J was the problem. How to resolve the problem was another matter. I ended up creating a custom routine for logging errors. When I deployed it on the server, however, I encountered the same problem as before, no response and no logs. For the second time, I picked up the phone and called GoDaddy. Ten minutes later I understood the whole problem. Java applications did not have permission to write to the file system!

In retrospect, this is fairly obvious since all users on the shared server run under the same Java virtual machine instance. Tomcat and Java just do not have the fine-grained ability to assigned directory-based permissions in such as configuration.

Of course, this meant that Log4J had to be directed to a different logging facility. The help desk guy suggested the /tmp folder, which Java can write to. I declined, as this was a public folder. I ended up using a JDBC appender and writing the log file messages to the MySql database. Of course, this led to the creation of an administrative page for viewing the log messages. I also ended up directing some log messages to the /tmp folder, specifically ones involving events occurring before database connectivity.

Finally, I had the web application working on GoDaddy. A few days later, after more development, I created a new build and deleted the old one. Or, at least, I tried to delete the old one. Some unknown process still had a lock on most of the files in the web application directory. I waited until Tomcat bounced and tried again. No luck.

For a third time, I picked up the phone and called GoDaddy support and they were able to delete the files for me. When I asked why I had trouble deleting the files, they informed me it was a 'glitch in the system' (Translation: 'We have no idea.').

What I've concluded is that accessing a web application creates file locks until Tomcat bounces. Therefore, if you delete the web application folder first thing in the morning, it will go quietly. If you access the web application, you're stuck with it until the next bounce. This led to the creation of a new build/deployment process.

My builds now use version numbers in the WAR file name. For example, instead of using 'app.war' I use 'app001.war' as the file name. This creates an /app001 folder on the server. The next deployment is named 'app002.war' resulting in an /app002 folder. I simply wait 24 hours after deploying a new build before I try to delete the old one.

This led to SEO and other problems. Since, I wanted my users to land on the web application's main page, I needed to redirect them from the server root location to a new folder such as /app123. The obvious choice is mod_rewrite, as it allows the redirection to be completely transparent to the user and to search engines crawling the site. Naturally, I have not been able to get this to work. A single call to GoDaddy did not fix the issue (I was told to wait 24 hours after placing the .htaccess file on the server).

Unable to use mod_rewrite, I turned to PHP. Currently, I am using an index.php to issue a 'HTTP/1.1 301 Moved Permanently' to the client. I am not sure how the search engines will like being redirected to /app123 one week and /app124, the next. I hope to get mod_rewrite working with GoDaddy's help in the near future and toss out the PHP file.

Outside of the joys of deployment, I have also discovered a few interesting tidbits about GoDaddy's MySql installation. GoDaddy is currently using version 4.0.24 of MySql. This version lacks support for a few really neat

things, like BOOLEANs, TRUE/FALSE keywords, and QUERY EXPANSION. Therefore, I recommend installing the same version on your development workstation. (When I started development, I mistakenly assumed that the SQL syntax differences would be trivial enough that I could get away with using MySql 5.0. A bad decision that I hope you will not repeat.)

I'm sure there are more GoDaddy Gotchas that I will learn about as I complete development. For instance, I foresee issues with browser session timeouts. The GoDaddy support pages do not specify what the default timeout is or how it might be changed. My guess is that it is global for all shared hosts and fixed.

Despite all the twists and turns and potholes I've encountered, I'm happy with GoDaddy in an overall sense, especially considering the price. The most important lesson I've learned is to allocate more time than you may expect.

- PAD File Manager
- Free Press Releases

About the Author

-   **Russell Thackston**
  I am a veteran of the U.S. Air Force and I have a BBA/MIS degree from Georgia Southern University and an MS/CIS from Georgia State University.

Related Web Design Articles

- Applications can really be developed with Ruby on Rails
- Cyberspace Communication & Forums
- Why I Hate Technology...
- Bandwidth and Data Transfer
- Using the HTTP_USER_AGENT

Article Categories

- Marketing - (61)
- Distribution - (4)
- Development - (13)
- Web Design - (18)
- SEO - (27)
- E-Commerce - (2)
- Customer Service - (5)
- OISV Ideas And Updates - (1)
- Running Your Business - (1)

Thanks to our Charter Sponsors

Search: [   ] Go!

OISV Home | Articles | Forums | About Us | Contact Us | Join OISV | Legal Notices
© Organization of Independent Software Vendors | Microsoft is a registered trademark of Microsoft Corporation