Development-Direct Connect

Overview

By default, both Xcode and Eclipse/WOLips will configure your application to use Direct Connect. When your WebObjects application launches, it listens on a particular port (assigned by the WOPort configuration option). Under a full-blown deployment scenario, this port is used by the mod_WebObjects adaptor or the WebObjects cgi-bin adaptor to communicate between Apache and your application. However, it's also possible to run in development completely on the Direct Connect port (i.e. without a webserver). In this configuration, one noteable difference is that all resources are served by the Direct Connect server as opposed to the Apache scenario where resources are served by Apache directly and your WebObjects application only handles application requests.

wotaskd relies on direct connect being enabled so that the adaptor can communicate with it. Do not globally disable direct connect on your servers. For example, if you disable direct connect from the WebObjects.properties file in the application server's home directory, you will prevent wotaskd from responding to requests on its assigned port.

Turning Off Direct Connect

- Configure your Apache to use mod_WebObjects or the cgi-bin adaptor
- Ensure wotaskd is running
- Set WODirectConnectEnabled to false
- Set WOAllowRapidTurnaroundMode to false (It doesn't mean what you think it does - only useful pre-Eclipse days) - if you don't disable this then your static resources will still be served by the Java app instance and not by Apache.
- Setup your WebServerResources folder so that it is reachable by Apache, eg by making a symbolic link from the folder CGI-Executables /WebObjects to MyGreatApp.woa in your build folder in Eclipse. Note: You may need to experiment with disabling the incremental builder setting to get build products that you can symbolic link to from the Apache root directories.
- make sure that in your wotaskd. settings your WOHost settings is the same as in Eclipse (by default there is no -WOHost setting in the run configuration in Eclipse, you have to add that). -WOHost localhost is a nice one.
- Profit

Logan Allred

I've heard all the comments about Direct Connect being evil, and I've been bit many times by its quirks, but I'd like to know from those that eschew Direct Connect what their workflow is and how they benefit (especially if they use Eclipse/WOLips).

I still use Direct Connect, mostly because I'm used to the workflow and have figured out most of the things that can go wrong by now. I like the fact that all of my development resources are all in one project and it seems I get some niceties in WOBuilder when using Direct Connect.

Maybe it's because most my projects focus mostly on data and not presentation (I have only a few graphics and an occasional stylesheet--though I'm beginning to use stylesheets more), and that I go back and forth between developing on Windows and Mac (so scripting can become a pain). My apps have not yet suffered because of the speed of loading graphics (far more the speed of database access).

It's not that I fear web servers, I'm just lazy 🙂 and not used to a workflow where I have "split development" instead of "split deployment". How do you version and manage your web resources being in a completely different location than your development resources (do you use resource folders in Eclipse, or separate projects? or some other nice Eclipse feature I haven't learned about? manual copying?). How do you handle deployment across development /staging/production? How much do you have to customize or modify your projects?

I can definitely see on sites with heavy presentation or dedicated graphic designers where using the web server would be a real plus, but I'm usually on a small team of 2-4 developers with minimal presentation.

I'm sure there's a better workflow than I'm using, I'm curious what everyone else considers to be development and deployment best practices for WO. What am I missing out on?

Chuck Hill

First off, it is not just about images. It is about having the power of a real webserver and about avoiding problems that only appear when you move to running the app through a real web server. Think of things like:

- caching, mod_expires is NOT your friend
- HTTPS
- SSI
- Rewrite rules
- headers only present when going through a web server

I'm sure you can think of more differences. I'd rather develop with all the services I will get in deployment and also find the problems that crop up *before* the app has been in production for half a day! I prefer my pain to be at leisure, not in panic. And, at least on OSX, it is dead easy to get Apache going.

As for work flow, what work flow? Our "projects", as seen from the source repository, are a hierarchy of folders. One is for code and that is where the WO projects go. Another is DocumentRoot and that is where, well, the document root stuff goes. Other folders are for docs, design, tests, etc. It is easy enough to check each one out where it goes. If your images are mostly static, so much less to do. If they change a lot, then the designers etc. can check in and out what they need in a way very familiar to them without needing to puzzle over arcane things like an Eclipse project. Image updates to your local doc root happen the same way that code updates to your workspace does.

For deployment, we have scripts that do command line Ant builds of all the code,tar up all the various resources (doc root included), and just generally make an install package. Another script on the target server takes that package, untars it, and copies the bits where they need to go. I don't understand how having to do split installs makes this any easier.