

Setting Up a New OS X Development Environment for ColdFusion

Matt Woodward (matt@mattwoodward.com)
February 14, 2007

Since I'm the proud owner of a brand new MacBook Pro, I thought while I'm doing the initial setup I'd outline what I install on a machine that I'm going to be using for day-to-day development tasks. Obviously everything I outline here is my own personal preference, but I get questions about doing development on a Mac more and more these days, so hopefully this will at least give people some guidelines.

After doing the requisite system updates to make sure the OS itself is up to date, I typically install the following applications on a fresh Mac.

1. Firefox and Extensions

Mac fanboys can hate me all they want for this, but Safari is a crappy browser. Use it once and once only to download Firefox (<http://www.mozilla.com>). Once you have Firefox installed, start grabbing up all the extension goodness.

- All-In-One Sidebar (<https://addons.mozilla.org/firefox/1027/>)
Puts all your Firefox "stuff" (bookmarks, history, downloads, extensions) in one place
- DOM Inspector (<https://addons.mozilla.org/firefox/1806/>)
As the name indicates, allows you to easily inspect the DOM of any web page
- Download Status Bar (<https://addons.mozilla.org/firefox/26/>)
Displays download progress for each download file in the status bar
- Down Them All! (<https://addons.mozilla.org/firefox/201/>)
Excellent download accelerator that also includes dTaOneClick, which allows you to easily pull down files such as images, MP3s, etc. from web pages.
- Fasterfox (<https://addons.mozilla.org/firefox/1269/>)
Gives you granular access to networking settings for things like simultaneous connections, cache, DNS cache, and more
- Firebug (<https://addons.mozilla.org/firefox/1843/>)
Insanely great extension for web development that lets you monitor practically any aspect of a page "live," including JavaScript and CSS. If you're a developer, you need this extension.
- FireFTP (<https://addons.mozilla.org/firefox/684/>)
Full-blown FTP client (and a darn nice one at that) that runs right inside Firefox. I've been using this as my main FTP client on my Windows box for quite some time since there's nothing as nice as the Mac FTP client Transmit (<http://www.panic.com/transmit/>) for Windows. It's extremely slick and shows just how much you can do with XUL (<http://www.mozilla.org/projects/xul/>).
- Forecastfox (<https://addons.mozilla.org/firefox/398/>)
Not critical for development, but it's a nice little weather add-on that can help you see if you should "work from home" due to inclement weather.
- Google Toolbar (<http://www.google.com/tools/firefox/toolbar/FT3/intl/en/index.html>)
Lots of handy Google goodness all in one nifty toolbar
- Mouse Gestures (<https://addons.mozilla.org/firefox/39/>)
I first got addicted to mouse gestures when I was using Opera (<http://www.opera.com>) full-time for a while (and Opera is still a vastly superior browser to Safari), and although

I don't use them as much anymore, they're still pretty slick. Basically you can do all sorts of browser control and navigation by making simple gestures with your mouse. For example, right-click and an "up" gesture might open a new tab, and right-click "L" could close a tab. Navigating back and forth with a simple mouse click is pretty cool.

- Pearl Crescent Page Saver Basic (<http://pearlcrescent.com/products/pagesaver/>)
Allows you to take screenshots of either entire web pages (meaning including the stuff you'd have to scroll to see!) or the visible portions of web pages.
- Sage (<http://sage.mozdev.org/>)
Slick little RSS and Atom feed reader
- Web Developer (<https://addons.mozilla.org/firefox/60/>)
More handy tools for web developers

2. Eclipse 3.2.1 (<http://www.eclipse.org>)

Eclipse is of course the development IDE on which you'll run CFEclipse and Flex Builder. Yes, you want to run CFEclipse as opposed to any other IDE you think you like. Trust me.

Installation of Eclipse is easy—download the .tar.gz file, extract it by double-clicking, and then drag the eclipse directory into your Applications folder (at least that's what I do). Launch Eclipse, choose a location for your default workspace (I just accept the default), then drag the Eclipse icon on your dock somewhere you want it to be permanently. By dragging it somewhere it will stay on the dock for easy access after you close Eclipse (which is rare for me).

Note that Eclipse is a tremendously powerful IDE platform for which there are loads and loads of plugins available. If you have a need for a tool to run within Eclipse, chances are someone's built a plugin for that purpose. I'm outlining the ones I use most often here, but a simple Google search or visit to Eclipse Plugin Central (<http://www.eclipseplugincentral.com/>) will help you find countless other plugins to help you reach IDE nirvana.

3. CFEclipse (<http://www.cfecclipse.org>)

CFEclipse is a fantastic free CF editor that's more or less single-handedly maintained by Mark Drew (<http://www.markdrew.co.uk/blog/>). Version 1.3 just came out recently (as of the date of this writing), and aside from a bug with cfscript parsing it's 110% stable. Just leave cfscript parsing turned off until a patch comes out.

While you can do a manual installation of CFEclipse if you like, Eclipse has a fantastic update mechanism built right into the IDE, so it's quicker to follow the instructions here (<http://cfecclipse.org/index.cfm?event=page&page=download>) to get CFEclipse installed. Launch Eclipse, go through those steps, and you'll have CFEclipse up and running.

After installing CFEclipse (and performing the recommended restart of Eclipse), go to Window/Open Perspective/Other in the Eclipse menu. From the Open Perspective menu choose CFEclipse, and this will put you in the CFEclipse perspective within Eclipse. If that all works, the installation was successful.

For more info on using CFEclipse (and Eclipse in general), check out Mark's excellent CFEclipse TV screencasts (<http://cfecclipse.org/index.cfm?event=page&page=TV>).

4. Subclipse (<http://subclipse.tigris.org/>)

Since we're all good developers and always use source control (and if you don't, shame on you—use this as your opportunity to start), and since Subversion (<http://subversion.tigris.org>) is the best source control out there, we'll need a way to access SVN from Eclipse. There are other plugins out there, but my preferred SVN plugin for Eclipse is Subclipse.

As mentioned above, Eclipse has a great update mechanism that allows you to easily install plugins that have an update site, and Subclipse is one of these type of plugins. Just follow the instructions here (<http://subclipse.tigris.org/install.html>) and you'll be up and running.

Note that this does NOT install Subversion on your Mac; this only gives you a way to access SVN repositories from within Eclipse, which is far more handy than using an external tool to do so. That being said, when things get wacky as they sometimes do, it's good to have an alternative means of accessing SVN, so we'll look at svnX in a moment. First we have more Eclipse plugins to install.

As another aside, for more information about using SVN, be sure and check out the excellent, free book “Version Control with Subversion” (<http://svnbook.red-bean.com/>). If you don't mind spending a few bucks, I also highly recommend “Pragmatic Version Control Using Subversion” (<http://www.pragmaticprogrammer.com/titles/svn/>).

5. XMLBuddy (<http://www.xmlbuddy.com>)

XMLBuddy is a great, free XML plugin for Eclipse. It's not the most capable XML plugin in the world, but it does the job quite nicely and is free. It supports XML, DTD, and XML Schema very well. If you need more XML capabilities in Eclipse, check out oXygen (<http://www.oxygenxml.com/>) or XMLSpy (http://www.altova.com/features_eclipse.html), but I'd recommend trying XMLBuddy first and if you find it doesn't meet your needs, then try some of the commercial XML plugins.

One weird thing about XMLBuddy is that their web site is always out of date. The near-latest version is available at download.com, but I found the latest latest (version 2.0.9) version at softpedia.com (<http://www.softpedia.com/get/Internet/WEB-Design/HTML-Editors/XMLBuddy.shtml>). Note that XMLBuddy does not work on Vista, which is yet another reason to get a Mac.

After downloading and extracting the zip file, you'll see a `com.objfac.xmleditor_2.0.9` directory on your desktop. Open a finder window and navigate to your `eclipse/plugins` directory (which if you put the eclipse directory in Applications will be under `/Applications/eclipse/plugins`). Drag the `com.objfac.xmleditor_2.0.9` directory (the whole directory, not just the contents of the directory) into this `plugins` directory and restart Eclipse if it's running. In Eclipse go to the Window/Preferences menu and if you see XMLBuddy listed on the left-hand side of the preference panel, you installed XMLBuddy correctly.

By default your XML files will automatically be associated with XMLBuddy inside Eclipse, but you can also right-click files and choose “open with” as needed. If you want to use XMLBuddy to edit things with other file extensions, simply go to Window/Preferences in Eclipse again, then expand the “General” section, then expand “Editors,” then click on “File Associations.”

From there, choose the extension you want to allow XMLBuddy to edit and add XMLBuddy as an editor. You can have multiple editors associated with file types, so you don't necessarily need to eliminate the editor that may already be there. For example, if you have something like a Mach-II configuration file that has a .xml.cfm extension, you can add XMLBuddy as an editor for .cfm files, then right click the file in question and choose "open with" to use XMLBuddy for that file. Once you open a file with an editor other than the default, Eclipse remembers this and will automatically use this editor next time you open the file.

6. DBEdit (<http://sourceforge.net/projects/dbedit>)

DBEdit is another fantastic plugin for Eclipse that lets you browse and interact with any database that has a JDBC driver right from within Eclipse. If you haven't tried this mode of developing before, you'll be amazed at the productivity you can gain from being able to access everything you need from within your IDE. Be aware that DBEdit is not a replacement for the full-blown management tools for your database of choice (e.g. SQL Server Enterprise Manager), but for the majority of basic database work that you do while developing this plugin does the trick.

After downloading the zip of the distribution (which as of the time of this writing is `dbedit_1.0.3_1.bin.dist_3.X.zip`), unzip the file to your desktop. Open that folder, and inside there you'll see two directories, `features` and `plugins`. Installing DBEdit is a manual process similar to XMLBuddy, but in this case there are both features and plugins you need to install as opposed to it being a simple plugin.

Navigate into the `features` directory and you'll see the directory `dbedit_1.0.3`. Drag that entire directory (again, the directory itself, not the contents of the directory) to the `features` directory of your Eclipse installation (e.g. `/Applications/eclipse/features`). Follow a similar process for the `plugins` directory, inside which you'll see four subdirectories. Drag all four of these directories to your Eclipse `plugins` directory.

Restart Eclipse if it's still running. To test your installation, go to Window/Open Perspective/Other, then choose DBEdit and click OK. If DBEdit is in the list and you don't get any errors by selecting it as your perspective, the basic installation is done.

The neat thing about DBEdit is that it can connect to any database that has an available JDBC driver. Since the most common databases used by CF developers are SQL Server and MySQL, I'll cover how to install the drivers for and connect to these databases.

Stephen Collins, who is responsible for the amazing ACME Guide (see below for more details, but you can find the ACME Guide at <http://www.acidlabs.org/extras/acme/>), pointed out in a blog post (<http://blog.acidlabs.org/2007/02/16/dont-forget-the-drivers/>) that in my description of where to put the drivers, this does not address using them with ColdFusion. This is a very good point. What I'm describing below works with DBEdit because you point directly to the database driver file when you do the setup.

In order to get these drivers working with ColdFusion, you need to add the jar file to ColdFusion's classpath, or you can drop the jar file somewhere that's already in ColdFusion's classpath. By default, depending on your installation type, ColdFusion will look in `{CF_ROOT}/lib` for a standalone installation (I hope that's correct; I always do the multi-server installation so I don't have access to see if this is right), or in a multi-server

installation it will look in `{CF_ROOT}/servers/lib`. At least it's supposed to look in `{CF_ROOT}/servers/lib` in a multi-server installation, and if that does work, that driver will be available to all your CF instances. Personally, I've had trouble with this in some environments (OS X and Linux specifically), so if you drop the jar file in that directory and CF doesn't seem to be able to find it (i.e. you get the "no suitable driver" error when attempting to configure your datasource), you may have to drop the jar file in the lib directory for the specific CF instance and bounce (restart) that instance. The lib directory specific to each CF instance is:

`{CF_ROOT}/servers/{SERVER_NAME}/cfusion-ear/cfusion-war/WEB-INF/lib`

If you have problems, drop the jar file there, restart that CF instance, and you should be good to go.

As another aside, if you want to add the jar file to the classpath manually to your CF instance, you'll have to log into the JRun administrator (N.B. that's the *JRun* administrator, *not* the ColdFusion administrator) and modify your classpath settings there. By default the JRun administrator lives at <http://localhost:8000/> When I tried to fire up the admin server from a terminal window on my Mac I get a weird "Java for ppc cannot run in this configuration" error, and I haven't looked into it further.

SQL Server (<http://www.microsoft.com/sql>)

For SQL Server, you'll need to download the free, open source jTDS driver (<http://jtds.sourceforge.net/>), which is an absolutely killer driver for SQL Server that you may even want to use in production as opposed to using the SQL Server driver that ships with CF (or worse, Microsoft's own SQL Server driver). This driver works fine with both SQL 2000 and SQL 2005 (both full version and Express). It may work with earlier versions of SQL Server as well but I've never tried anything older than SQL 2000.

After downloading the zip file of the distribution (you don't need the source code unless you're really interested in it), unzip the file to your desktop. Since you will be pointing directly to the jar file for the driver it doesn't really matter where you put the files. I created a directory called `database_drivers` in my Documents directory and put the `jtds-1.2-dist` directory in there, but you can put them where you want.

The next step is to set up a connection. (This step assumes you have an available SQL Server you can hit.) In Eclipse, go to the DBEdit perspective if you aren't already there, and click "Connection" at the top, then click "Configure." Under the Favorite Connections panel on the left, click "New." Name the bookmark whatever you want (I called mine "test"), and then click the Classpath button above the configuration panel on the right. Click the "Add Archive" button and navigate to the `jtds-1.2.jar` file, then click OK.

Click on the "Common" button at the top of the right-hand panel again, and you should now be able to select `net.sourceforge.jtds.jdbc.Driver` from the JDBC Driver drop-down. If you can't, double-check the settings from the paragraph above.

The server URL follows a JDBC URL pattern that is driver-specific, so in the case of the jTDS driver for SQL Server, the URL pattern is as follows:

```
jdbc:jtds:sqlserver://SERVER_NAME_OR_IP:PORT/DATABASE_NAME
```

There's other stuff you can put in this URL, but that should be sufficient for most situations.

Enter the user name for the database you're connecting to, check the box next to "save password" (unless you want to be prompted every time you connect), type in the password for the database, and hit "apply" to save the bookmark for the connection. Then click "Connect" in the lower right-hand corner of the Connection Configuration dialog box, and you should see your database connection appear in the left panel of the main Eclipse perspective window. From there you can start navigating through your database and playing around with all that DBEdit has to offer.

MySQL (<http://www.mysql.com>)

For MySQL the basic instructions follow the SQL Server instructions outlined above, you'll just need different JDBC drivers. Download the latest JDBC drivers for MySQL from the MySQL web site (<http://dev.mysql.com/downloads/connector/j/5.0.html>). As of the time of this writing the latest drivers are version 5.0.4. (Note that if you follow the "Definitive Guide" that is linked to below, which outlines configuring CF 7.0.x to work with Apache on OS X, it recommends that you use version 3.1.x of the MySQL Driver. This earlier driver has a bug, specifically on tinyint fields, whereby 1 comes back as 257 and 0 comes back as 256. I've had absolutely no problems with the newer driver, and it doesn't have this bug.)

As with the SQL Server drivers above, extract the contents of the zip file and then drag the contents of the zip file to a location of your choosing. Open the DBEdit connection window and follow the process outlined for SQL Server above to point to the jar file, which for MySQL is `mysql-connector-java-5.0.4-bin.jar`.

After pointing to the jar file you should be able to select `com.mysql.jdbc.Driver` from the JDBC Driver drop-down on the Common tab in DBEdit. The URL pattern for the MySQL JDBC driver is as follows:

```
jdbc:mysql://SERVER_NAME_OR_IP:PORT/DATABASE_NAME
```

Again enter the user name and password for the database and test the connection.

As you need to connect to different types of databases that use other JDBC drivers, Pete Freitag has a very handy guide to the various JDBC Driver URLs (http://www.petefreitag.com/articles/jdbc_urls/) that you'll want to bookmark.

7. svnX

(<http://www.lachoseinteractive.net/en/community/subversion/svnx/features/>)

As I mentioned above, it's extremely handy to have the means to get to SVN other than through Subclipse. The stand-alone client svnX is a fantastic tool and lets you do some things that are difficult or not available within Subclipse.

svnX does rely on underlying command-line SVN tools, so the first step in installing svnX is to install these command line tools. Go to the downloads page on the svnX web site (<http://www.lachoseinteractive.net/en/community/subversion/svnx/download/>) and download Martin Ott's binary package for SVN. Install this first. After that, go ahead and install svnX. You don't need to worry about playing with this for now, and since this isn't a guide to SVN in general, I won't discuss this tool further. You can refer to the SVN books mentioned above for more information on using SVN. At this point, just know that svnX is an extremely nice tool for

accessing SVN from OS X.

8. ColdFusion 7.0.x (<http://www.adobe.com/products/coldfusion>)

Installing ColdFusion on OS X isn't quite the pain it used to be, but with the advent of Intel-based Macs, it's not quite as simple as it used to be either. Download the developer edition of CF for OS X from [adobe.com](http://www.adobe.com), then follow the excellent "Definitive Guide" written by Mark Andrachek (<http://webmages.com/geek/cfm-x-on-intel-macs>). If you want to use ColdFusion's built-in web server you can just do a simple install, but for the ultimate dev environment I highly recommend going to the trouble to get things working exactly as Mark outlines.

Not a huge pain, but let's hope Scorpio is a bit more Intel-Mac friendly.

9. MySQL (<http://www.mysql.com>)

There are of course various databases available that will run on OS X, but my preferred database for OS X is MySQL. Download the latest Community version (<http://dev.mysql.com/downloads/mysql/5.0.html>) for OS X, which will be towards the bottom of the downloads page. Make sure and get the version appropriate for your OS version and your processor type. If you have a newer Mac you'll likely want the version for OS X 10.4 (x86). I always just download the standard version (not the Max version). The latest version as of the time of this writing is 5.0.27.

Download the file and expand the disk image. You'll see two .pkg installers in the disk image. First run the `mysql-standard-5.0.27-osx10.4-i686.pkg` file. For people familiar with (let's be honest) "friendlier" databases such as SQL Server, the MySQL installation will seem anti-climactic because it doesn't install any GUI tools.

Next, run the `MySQLStartupItem.pkg` installer. This installs an additional piece on OS X so MySQL will fire up automatically when the machine boots. Don't worry if this isn't what you want, because we'll look at how to control this in a moment. For now just run this installer.

Next, install the `MySQL.prefPane` file. This adds a MySQL control panel for starting and stopping MySQL to your System Preferences application. From this panel you'll be able to start and stop MySQL as well as tell OS X whether or not it should start MySQL on boot. After you install this preference panel it should launch, and you'll notice that MySQL is not running. Click the "Start MySQL Server" button and you should see the status change to "running."

Next let's test this installation. Open a terminal window (you'll learn to love the terminal on OS X over time ...) by going to Applications, then Utilities, then double-click Terminal. Add the Terminal to your dock permanently because you'll be using it a lot as you develop on your Mac.

(As an aside, I like to make my terminal the fancy "green on black" color scheme and give it a bit of transparency just to make it look cool. To make your terminal look as cool as mine does, click on the Terminal menu in Terminal, then Window Settings, then choose Color from the drop-down. Choose "Green on black" and increase the transparency a bit, then click "Use Settings as Defaults" so every terminal window you open will look just this cool.)

Back to checking the MySQL installation. In your terminal window, navigate to the MySQL bin directory (in your terminal type `cd /usr/local/mysql/bin` and hit return), then type the following command and hit return:

```
./mysql -u root
```

This puts you at a mysql command prompt. Type `show databases;` and hit return, and you should see a list of databases, which at this point consists of `information_schema`, `mysql`, and `test`. If you get this far without any errors, then MySQL is running fine.

For some additional friendliness in using MySQL (personally I'm a bit of a terminal gearhead where MySQL is concerned), you may want to download and install some of the GUI tools (<http://www.mysql.com/products/tools/>) such as the MySQL Administrator, Query Browser, and Workbench. NOTE: I personally have had the Administrator and Query Browser crash a lot on one of my machines, so I'm not currently using these tools. I'm sure MySQL is aware of the issues and is working on them. As with all of this stuff, YMMV, so feel free to give them a try if you're so inclined.

Alternatively, there are some nice MySQL GUI tools available that run natively on OS X. If you're interested in some of these tools, go to <http://www.versiontracker.com> and do a search on MySQL for a list of some options. I haven't used any of these tools in quite some time (I remember liking an app called SQL Grinder, and it just used JDBC drivers so you could use it to connect to SQL Server as well) so I won't be able to competently recommend any of them at this point.

10. TextWrangler (<http://www.barebones.com/products/textwrangler/>)

In addition to Eclipse, a nice, basic, powerful text editor is a must for a good development environment. BBEdit is a nice one (<http://www.bbedit.com>), but it costs money, and I don't use the vast majority of features that are available in BBEdit. Thankfully, BareBones Software makes a lighter version of BBEdit called TextWrangler available for free (<http://www.barebones.com/products/textwrangler/>). It's the Swiss Army Knife of text editors (at least where free editors are concerned) so download it and install it, because you'll need it at some point. I don't use it a lot but when you need it, you need it.

11. Parallels (<http://www.parallels.com>) and Windows XP Professional

Unfortunately there are some apps that developers need to use that require Windows. Whether it's SQL Server Enterprise Manager, or Visio, or other Windows-specific apps, sometimes you just need Windows. Sad, but true.

The good news is that you don't need a Windows box to run Windows apps. For the low, low price of \$79.99, you can run Windows in a virtual machine (VM) on OS X. Head over to <http://www.parallels.com> and give them your money. Seriously. You will NOT regret it. Of course you also need a valid Windows XP license that you are not running on another machine to run in the virtual machine, so make sure you're legit on that front. (You didn't hear this from me, but don't pay full retail for Windows. You can get Windows XP Professional OEM from newegg.com for cheap:

<http://www.newegg.com/Product/Product.asp?Item=N82E16832116059> Just be aware of the licensing restrictions around OEM copies of Windows.)

I won't go through the whole spiel on how to get a VM up and running, but it's extremely simple and at the end of the process, you'll have a nice XP VM running in a window on your Mac. Note that in the new version of Parallels that isn't out yet, you'll even be able to launch Windows apps from your OS X dock and run them outside the context of the VM window. VERY cool stuff.

I can't recommend running Windows Vista in a VM at this time. I successfully installed it but it's a bit unstable, and at one point the Vista VM even crashed OS X itself. XP runs beautifully (BEAUTIFULLY!) in Parallels, and since it's less resource intensive than Vista, it's a better choice provided the Windows apps you need to use run in XP (which they still do at this point since Vista is so new).

What do I run in Windows? Basically it's just SQL Server Enterprise Manager, Groove, and a scattering of other apps that I need occasionally such as Visio. It's just nice not to have to give up the Mac or have a separate physical box just to run these few items.

Note that the performance of the VM is FANTASTIC. If you've used VirtualPC on a Mac in the past, don't even think of this as the same animal. Parallels on an Intel Mac runs Windows at native speed (same underlying hardware as Windows machines, ergo a LOT less emulation than was previously required), so you will not be disappointed with this setup.

12. Other Resources

While it isn't Mac specific, I can't say enough good things about the ACME Guide (<http://acidlabs.org/extras/acme>) by Stephen Collins. The ACME Guide covers installing Apache, ColdFusion, MySQL, and Eclipse (hence the ACME acronym) and is an extremely focused and practical guide.

So there you have it, a semi-brief guide to how I typically set up a fresh Mac to use and a development environment for ColdFusion. I'm sure there are some holes, missteps, falsities, etc. in this since I was typing it pretty feverishly as I installed everything today, so please let me know if you run into any issues (matt@mattwoodward.com). All told you can get everything up and running in probably half a day or less since that's about how long it took me even while typing out this guide.

Hope that helps, and I hope it inspires some of you Windows users to jump ship before Vista makes you hate Windows all over again.