**Want a FREE WebSphere Eclipse IDE and development server with that?**

Posted by [Stuart Smith](http://blog.webagesolutions.com/archives/author/ssmith) in [Eclipse](http://blog.webagesolutions.com/archives/category/eclipse), [Java EE](http://blog.webagesolutions.com/archives/category/java/java-ee), [WebSphere](http://blog.webagesolutions.com/archives/category/websphere" \o "View all posts in WebSphere) on December 20, 2011

**UPDATE:** We’ve posted some Java EE 6 development courses that use WebSphere and Eclipse to highlight this capability.  The best place to find these classes is in the [WebSphere 8.0 Programming category](http://www.webagesolutions.com/webapp/wcs/stores/servlet/Category_10001_10051_15507_-1_10009,15505_websphere-application-server-8-0-programming-training" \t "_blank).

**UPDATE:** If you are also interested in Spring tools for WebSphere development you might check out a related blog post – [How about a FREE set of WebSphere and Spring Eclipse Development Tools?](http://blog.webagesolutions.com/archives/866)

**UPDATE:** Currently the versions of the Eclipse and WebSphere downloads are different than those detailed here.  The same process still works though and the important part is that if you want to run a WebSphere server from Eclipse (one of the main reasons to perform this setup) the Eclipse instance has to point to the IBM JVM. This is easiest to do by pointing it to the IBM JVM installed with the ‘WebSphere for Developers’ as outlined in this post.  Post questions if you have any as I try to keep up with the comments on this post.

**ORIGINAL POST:**

For the longest time the biggest complaint we’ve heard from developers who use WebSphere is “We can’t just use Eclipse!”.  In fact, the fact that the only “official” development support IBM would provide for WebSphere was Rational Application Developer created an entire market for [MyEclipse Blue](http://www.myeclipseide.com/blue.php" \t "_blank) of people that didn’t want to pay for RAD but developed for WebSphere.  Even MyEclipse Blue is not free though, it is just cheaper than RAD.

So to solve this problem you do not know how many WebSphere shops I’ve seen that:

* Do development with Eclipse
* Manually add Java EE/WebSphere JARs to the classpath
* Try to automate the process of exporting an EAR and deploying to WebSphere the best they can

Well now it is MUCH EASIER!  IBM has just released FREE Eclipse plugins to deploy/start/stop WebSphere v7.0 and v8.0 servers from the Eclipse ‘Servers’ view.  I bet you aren’t used to hearing IBM and “free” in the same sentence!  These are only available for WebSphere Application Server v7.0 & v8.0 (no WebSphere 6.1, WebSphere Portal or Process Server) but these are some of the most common WebSphere environments right now and in the near future.

All of this is done with the “WebSphere Application Server Developer Tools for Eclipse” in the “[WASdev](https://www.ibm.com/developerworks/mydeveloperworks/blogs/wasdev/entry/home?lang=en" \t "_blank)” community on IBM Developerworks.  Although you might see a lot of discussion about the WAS 8.5 “Liberty” profile which is an early program of some other exciting things they are doing, the WAS Developer Tools for Eclipse is something you can use NOW.

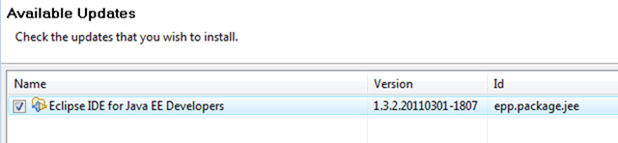
**Setup Steps**

Although you can certainly get some more info on how to use these new tools from the [official announcement](https://www.ibm.com/developerworks/mydeveloperworks/blogs/wasdev/entry/announcing_the_new_websphere_application_server_developer_tools_for_eclipse_v8_0_42?lang=en) and the WASdev [downloads](https://www.ibm.com/developerworks/mydeveloperworks/blogs/wasdev/entry/download?lang=en) page, I figure I would take it one step further and give you a complete set of instructions along with some screenshots for doing this on Windows.  It is really easy to do.

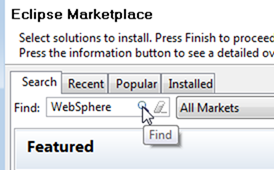
First of all I am going to assume for now that you are only using this for WebSphere v7.0 or v8.0 and not the “early program” (or Alpha) WAS 8.5 Liberty edition.  The reason this is important is the version of Eclipse to use.  For WAS 7 & 8 you will use Eclipse Helios.

**UPDATE:** It has been pointed out that there are 32-bit and 64-bit versions of both Eclipse and WebSphere.  The type you use for both has to be the same.  I would suggest the 32-bit since for development you don’t really need the expanded memory addresses of the 64-bit.  If for some reason you are using different architectures for the two products and getting an error probably the easiest way to fix it is to get the Eclipse version that matches the WebSphere you have installed.

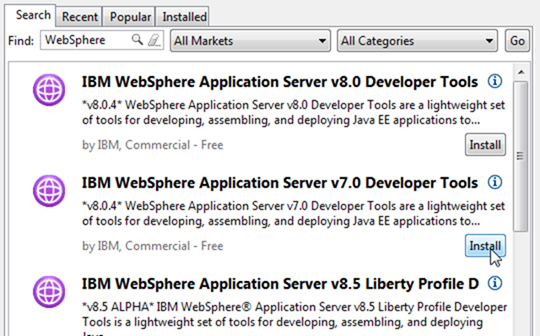
* First you will need to have a Java installation just like any other Eclipse environment even though we will just use it for the updates.  Later we will link Eclipse to the JVM installed with your local WebSphere server (this will be important).  You can get the Java download [here](http://www.oracle.com/technetwork/java/javase/downloads/index.html) although make sure it is not a ‘Java SE 7’ download as the Eclipse we will start with wouldn’t support that.
* Next, go get the Helios release of the [Eclipse for Java EE Developers](http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/heliossr2).
* Unzip the Eclipse download into an empty directory where you want Eclipse software to run from.  Even after we update with WebSphere tools it will run from this directory.
* Run the ‘eclipse.exe’ file in the root of where you extracted Eclipse and open Eclipse in an empty “workspace” directory.  We won’t be creating projects yet but we still need to open a workspace to run the updates we need.
* Make sure you are able to connect to the internet from within Eclipse.  If you have a proxy configuration this might mean going into the Eclipse preferences (**Window –> Preferences**) and the ‘**General –> Network Connections**’ section to setup network details.  A lot of times though Eclipse might be able to pick up your native internet settings so I would try this only if you have problems in the next steps.
* Close the preferences if you had them open.
* Once Eclipse opens, select ‘**Help –> Check for Updates**’ and wait for the update sites to be contacted.
* Leave the default selected to update ‘Eclipse IDE for Java EE Developers’ and click the **Next** button.

[](http://blog.webagesolutions.com/wp-content/uploads/2011/12/image.png)

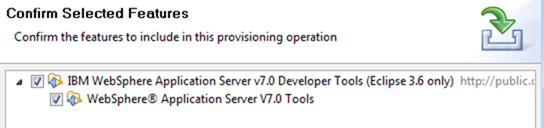
* Click the **Next** button again to confirm the update.
* Accept the license terms and click the **Finish** button.
* Once the installation of updates is complete, click the **Restart Now** button.
* When prompted, open to the same workspace as before.
* From the running Eclipse, select **Help –> Eclipse Marketplace**.
* On the dialog that appears, leave **Eclipse Marketplace** selected and click the **Next** button.
* On the Search tab that appears, fill in **‘WebSphere’** into the box and click the Find button.

[](http://blog.webagesolutions.com/wp-content/uploads/2011/12/image1.png)

* Select the WebSphere v7.0 or v8.0 tools depending on which you want and click the **Install** button next to the one you want.  Note that the WebSphere v7.0 tools have an 8.0.4 in the text description but this is just the version of the Eclipse plugins.  If you want to do both you can, just select one and continue and then repeat the process after restarting Eclipse after the first set of tools is installed.

[](http://blog.webagesolutions.com/wp-content/uploads/2011/12/image2.png)

* Leave the option to install your chosen tool version and click the **Next** button.

[](http://blog.webagesolutions.com/wp-content/uploads/2011/12/image3.png)

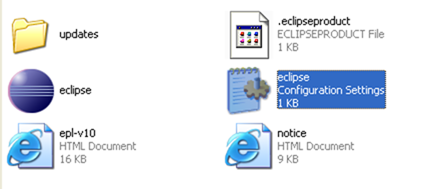
* Accept the license terms and click the **Finish** button.
* If prompted about installing unsigned content click the **OK** button to continue the installation.
* Once the installation of updates is complete, click the **Restart Now** button.
* When prompted, open to the same workspace as before.
* After Eclipse opens, exit from Eclipse.

So that’s all you need right?  Not quite.  Besides the Eclipse tools you will also need a local WebSphere 7 or 8 test server. I know some of you are saying “darn, I knew this couldn’t be completely free!” Fear not! I promised the development environment would be completely free and it is because IBM has offered for a while now the [WebSphere Application Server for Developers](http://www-01.ibm.com/software/webservers/appserv/developer/index.html" \t "_blank) product, a COMPLETELY FREE version of WAS for development that is NOT a trial or expiring edition and is built from the same codebase as the production versions. The best link I’ve found to go straight to the WAS for Developers downloads is [here](http://www.ibm.com/developerworks/downloads/ws/wasdevelopers/).  For those of you with IBM Passport Advantage download abilities there is also a ‘WebSphere Application Server for Developers – Eclipse Tools Edition’ which has all of this but there is also a lot of stuff you probably don’t need.

We will also need to link the Eclipse environment to the Java JVM installed with the WebSphere server.  If you don’t do this you won’t be able to define a WebSphere server in Eclipse as the WebSphere Developer Tools will check that you are using an IBM JVM.

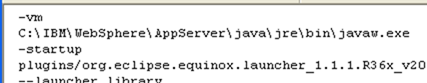
So just a few more steps and we will be set!

* After downloading WebSphere Application Server for Developers from the link above or [here](http://www.ibm.com/developerworks/downloads/ws/wasdevelopers/) go ahead and install it.  This will differ slightly depending on the version so it isn’t detailed here but should be pretty straightforward.  I always suggest NOT installing under the Windows ‘Program Files’ directories as on Windows Vista and above I have seen cases where Windows won’t let you save WebSphere configuration files so your configuration changes are mysteriously “lost”.
* Once you have WAS installed make sure you have a “profile” configured.  This is the configuration that will serve as your test server.  You can run the ‘Profile Management Tool’ if a profile isn’t created during installation.  If you run the Profile Management Tool I always like to select the “Advanced” profile creation because you can use the “development template” for better development performance, make sure the server uses the default ports and NOT have the server run as a Windows service.  No matter how it is created know the name of the profile that you will use.
* After WAS for Developers is installed go to the directory where you unzipped Eclipse.  Find the ‘eclipse.ini’ file and open it with a text editor.  The extension may be hidden and just list the file type as ‘Configuration Settings’.

[](http://blog.webagesolutions.com/wp-content/uploads/2011/12/image4.png)

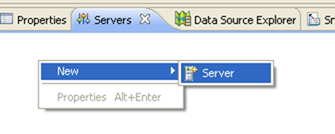
* Once you have the file open, add the following lines to the start of the file, substituting for <WAS Install Root> the directory you installed WAS at.  In the screenshot below my <WAS Install Root> was ‘C:\IBM\WebSphere\AppServer’.  Save the file when it is like below.  Make sure when you save it the file extension is not changed.

-vm  
<WAS Install Root>\java\jre\bin\javaw.exe

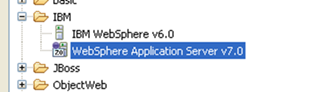
[](http://blog.webagesolutions.com/wp-content/uploads/2011/12/image5.png)

**UPDATE:** As suggested in the comments, another way to alter the JVM used to start Eclipse is to add the ‘-vm <Path to JVM>’ syntax to the ‘Target’ property of a desktop shortcut that is used to launch Eclipse.  Setting the value in the ‘eclipse.ini’ file is to set the default and the command line option overrides this.  Your approach will probably depend on if you develop WebSphere and non-WebSphere projects with the same Eclipse install.

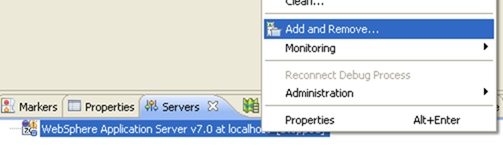
* Once you have modified and saved the ‘eclipse.ini’ file, open Eclipse again to the same workspace.
* Close the ‘Welcome’ page if it appears and find the ‘**Servers**’ view along the bottom.  This should open by default in the ‘Java EE’ perspective that should also be the default.
* Right click in an empty area and select ‘**New –> Server**’

[](http://blog.webagesolutions.com/wp-content/uploads/2011/12/image6.png)

* Expand IBM and select your version of WAS and click the **Next** button.  This is where you would have errors if you hadn’t started Eclipse with the IBM JVM.

[](http://blog.webagesolutions.com/wp-content/uploads/2011/12/image7.png)

* Since this is the first WebSphere server you defined, hit the **Browse** button and find the root directory of the WebSphere installation.
* Once you have the WebSphere installation directory click the **Next** button.
* On the next page make sure the correct profile name is listed in the drop-down.  Also make sure to supply security settings if you enabled security when the profile was created.
* Once your settings are configured click the **Finish** button.
* Start deploying projects from your workspace by right clicking the WebSphere server and selecting ‘**Add and Remove..**’, start and stop the server, even open the WebSphere Admin Console by going into the ‘Administration’ menu when right clicking!  In short, do all the things you’ve ever wanted to do to deploy and test on WebSphere from Eclipse!

[](http://blog.webagesolutions.com/wp-content/uploads/2011/12/image8.png)

**Conclusion**

So now you have a way to do WebSphere development with completely free tools without having to deploy remotely, hack the project classpath, come up with Ant scripts just to export and deploy, etc.  Obviously this opens up a whole new realm of possibilities for WebSphere development.  You will have much more freedom to construct the development environment that suits your project best but still make sure you are testing against WebSphere.

Since this is obviously a thing we know our own clients have been looking to do for quite a long time we plan on using this in several areas.  We will likely release a few Java EE training classes that use Eclipse and WebSphere 7.0 just to prove this is a completely capable development environment and let people get familiar with it.  We will also start using this quite heavily in the Java EE 6 on WebSphere 8.0 courses that will be posted soon.  We will also use this in our Spring 3 classes since one of our unique offerings has always been offering Spring development training on WebSphere.  Look for another blog post soon about the best development environment to use for Spring development for WebSphere.

So start dancing in the streets, we can finally develop and test WebSphere applications on “regular” Eclipse!

- See more at: http://blog.webagesolutions.com/archives/538#sthash.WEwjKP4d.dpuf