MyEclipse

Introduction

This is intended to lead you from scratch (well, you have Eclipse and MyEclipse installed and can look at our cvs repository) to obtain, configure, build and run applications in the glast-ground project.

All the projects currently live in the glast-ground directory under infrastructure in the cvs repository.

Getting the Project

Open the cvs repository view and navigate to the infrastructure directory. You'll see glast-ground under it. Right click on glast-ground and check it out. This will set up automatically as a project in Eclipse.



Once you have checked out the project you should see something like this:



Access to Needed Jar Files

You need to run the ant setup target to copy jar files to their appropriate home in classpath. Do this by right-clicking on setup in the Outline view and selecting Run->Ant Build.



These will leave you two jar files short.

Add them to the Java build path by right-clicking on the grits project and selecting Properties, then Java Build Path. Add junit and mockobjects from lib /build.

Info	Java Build Path				
CVS Java Build Path Java Compiler	Source Brojects Libraries Order and Export JARs and class folders on the build path:				
Javadoc Location	cglib-full-2.0.1.jar - grits/src/webapp/WEB-INF/i Add JARs				
MyEclipse-Web	Commons-collections-2.1.jar - grits/src/webapp/\ Gommons-lang-2.0.jar - grits/src/webapp/\/EB-It Add External JARs				
MyEdipse-XDoclet	commons-logging-1.0.3.jar - grits/src/webapp/W dom4j-1.4.jar - grits/src/webapp/WEB-INF/lib				
	ehcache-0.7.jar - grits/src/webapp/WEB-INF/lib Add Library				
	fillernate2.jar - grits/src/webapp/web-ivr/iib Add Class Folder				
	Iog4j-1.2.8.jar - grits/src/webapp/WEB-INF/lib mockobjects-core-0.09.jar - grits/lib/build Edit Edit				
	Image: Site of the second state of				
	arits/src/webapn/WEB-INE/classes				
	grits/src/webdpp/web-tive/classes				

If you didn't get them in the right place, they are generally to be found in the lib/runtime directory and should be copied into the afore-mentioned lib directory.

Q. -

in

Configuring to Build: Ant

Basically you'll want to tell Eclipse to use your workspace to build into. You get there by clicking on External Tools... option from the Run tool the tool bar to see:

San External	Fools						
Create, manage, and run configurations							
Create a confi	iguration that will	run an Ant buildfile.					
Configurations:		Name: glast-ground					
∦ Ant Build 		Main Refresh Build Targets Classpath Properties Model Buildfile: \${workspace_loc:/glast-ground/build.xml}					
		Browse Workspace Browse File System Variables					
		Base Directory:					
		\${workspace_loc:/glast-ground}					
		Browse Workspace Browse File System Variables					
		Arguments:					
		Variables Note: Enclose an argument containing spaces using double-quotes (").					
		Capture output					
New	Delete	Apply Revert					
		Run Close					

Next you'll want to set up your build targets, in order. Click the names you see in the bottom box that echoes the targets - in that order.

External Tools							
Create, manage, and run configurations							
Create a confi	iguration that will i	run an Ant buildfile.		D			
Configurations:		Name: glast-ground					
三	ild st-ground m	Main & Refresh & Refresh & Good Check targets to execute:	📆 Build 🛛 👫 Targets 🛛 🍫 Classpath 🛛 <🗈 Properties 🗎 🔜 _	• •			
		Name	Description				
		all (default target)	Run the tests and build the war file				
		ean 🖉 🖲 dean	Clean all built artifacts except generated source files and di				
		Compile	Compile all Java source files (regular source, generated so				
		Config	Asemble configuration files				
			Clean all built artifacts				
		bhm2iava	Cenerate java from hhm files				
		☑ Implify average in the second s	Scherate gave non monthiest				
		5 out of 17 selected ✓ Sort targets Target execution order: <pre> dean, xdoclet.hibernate, hibernate.properties, compile, config </pre> Order					
New	Delete		Apply Revert				
			Run Close				

Matt: why isn't ant set up in the repository version?

Now you can run the build by hitting the run button, which builds glast-ground by default. After you build your project, you should also recompile your JSPs, via a right-click on the project following MyEclipse->Recompile All JSPs.

J2EE server

We're using Tomcat 5 as the server. You can get it from http://jakarta.apache.org/tomcat/. Install it so you can have a local server to test with. Set it to port 8080.

₿<u>*</u> -

Next you'll need to tell Eclipse that you're using Tomcat. I think it just figures out you have Tomcat installed and you can start it from

Once it is started, you can deploy the project to the server via the toolbar button

JUnit tests

• pulldown menu. You can run the JUnit tests from the run button

Matt: did you do anything to configure my MyEclips to see the JUnit tests?

Hopefully you'll see something like this in the JUnit tab in the bottom right of the IDE

Problems	Tasks	Properties	Web Browser	Image Preview	Console	JUnit 🕅	수 🕆 🔳 🔍 🚮 👻 🗖		
Finished after 8.994 seconds									
Runs:	21/21		Errors: 0	🖾 F	ailures:	0			
∎ [⊠] Failur	es 📑	Hierarchy				≡ Failure Trace			
÷ E: ji	unit.fra	mework.Tes	tSuite						