Using the Swordfish Git repository with Eclipse

Motivation

Quite a few of the current contributors to the Swordfish project do not have committer status in the Swordfish Eclipse project. It was important to find a way to let them contribute in a controlled way in a repository that is available to the public. This repository should be

- writable for anyone who want to collaborate
- easy to synchronize with the official Eclipse SVN repository
- reliably about the creation of patches (we had problems with patches created / applied with Eclipse based on the SVN repository)
- allow for a CI build to ensure consistent & stable patches

Solution

The idea is to use a free public repository hosted at github.com that is mirrored from the Eclipse SVN. Collaborators can either fork this and create patches or (in case of Sopera employees and partners) get direct access to the mirror repository.



Prerequisites

To give a short checklist, this is what you need to use Git:

- The Git command line tools
- The EGit Eclipse plugin
- An account with a public key at http://www.github.com
- To be added as collaborator at Olivers repository at http://github.com/owolf/eclipse-swordfish.git

How to use Git

In a blog post on http://swordfishing.wordpress.com we saw how to access the Swordfish Git repository using Git from command line. Now that may not be everybody's cup of tea. Some people (including me) are not so happy when forced to do **everything** from command line. I don't like it with CVS or SVN either.

Fortunately there is the "EGit" plugin for Eclipse (http://github.com/guides/using-the-egit-eclipse-plugin-with-github) – including a nice tutorial. The plugin is still in an early stage, so we'll need to install the command line tools anyway.

The next thing we need is an account on Github.com. Most probably we'll want to add a SSH public key to our account.

Finally Oliver to add we as collaborator (you'll need that to be able to push our updates to Olivers repository).

Now we can import the Swordfish Git (located at git://github.com/owolf/eclipse-swordfish.git), we can just import it into Eclipse using "Import ..."



Then we select Git as source and enter the Swordfish Git URI (located at git://github.com/owolf/eclipse-swordfish.git) as source.

000	Import Git Repository			
Source Git Repository Enter the location of the source repository.				
Location				
URI:	git://github.com/owolf/eclipse-swordfish.git			
Host:	github.com			
Repository path:	/owolf/eclipse-swordfish.git			
Connection				
Protocol: git	\$			
Port:				
Authentication				
User:				
Password:				
0	< Back Next > Cancel F	inish		

We press the "Next >" button.

000	Import Git Repository	
Source Git Reposito	pry	CIT
Select branches to cl		
		4
Branches of git://gith	nub.com/owolf/eclipse-swordfish.git:	
🗹 master		
Select All	Deselect All	
2	<pre>< Back Next > Cancel</pre>	Finish
U.		

Select destination and initial branch and press finish.

000	Import Git Repository	
Local Destination Configure the loo	on cal storage location for eclipse-swordfish.	GIT
Destination		
Directory:	/Users/jkindler/Documents/workspace_swordfish_olli/e	Browse
Initial branch:	master	*)
Configuration		
Remote name:	origin	
0	<pre>< Back Next > Finish</pre>	Cancel

Sooner or later the repository cloning will finish ...

Cloning from	git://github.co	m/owolf/eclipse	-swordfish.git	
Initializing local re	pository			
Receiving objects: 6%	(468/7786)			
Always run in backgrou	ind			
	Cancel	Details >>	Run in Background	1.

Then we need to go to a terminal window to create the Eclipse projects and import them as existing projects (as shown in Renats post at http://swordfishing.wordpress.com/2009/02/03/first-steps/).

Afterwards it's easiest to create an Eclipse project file in the root project (just copy the one from the API project, change the name inside), import it and share it as a Git repository. To do that you right-click on the eclipse-swordfish project and choose "Team / Share Project..." :

000	Share Project	
Share Project Select the reposito	ry plug-in that will be used to share the selected project.	
Select a repository	type: 	
0	< Back Next > Cancel (Finish

Press "Finish" ...

000	Configure Git Repository	
Configure Git Re Select Git Reposit	epository tory Location	GIT
Repository Locat	tion	
 Search for ex Create a new ✓ Create reposi 	isting Git repositories Git repository for this project itory in project's parent directory	
0	<pre>< Back Next > Cancel</pre>	Finish

Now the package explorer indicates that we are using Git:

📲 Package Explorer 🖾 🦹 Hierarchy	🖻 客 📴		3
swordfish [Git @ master]			
🕨 📂 org.eclipse.swordfish.api		- 1	

Some things will be a bit different from working with CVS or SVN. In these systems we are clearly a client who just received a working copy of the central repository. When we imported from the remote repository, we have created a <u>standalone</u> local repository of Olivers remote repository residing on github.com. So working with that requires slightly different steps.

To make it a bit more colourful, let's have a look at some typical scenarios:

- committing a non-conflicting fix
- committing a fix that causes conflicts
- resetting the local repository clone to the state of its origin

Committing a non-conflicting fix

Let's make a tiny change on the FilterStrategy interface:



After saving the change it has to be committed to the *local* Git repository first. So we choose the Team/Commit on the eclipse-swordfish root project:

000	Commit Changes
Commit Message:	
Added a GIT test comment!	
Author:	
Amend previous commit	
Add Signed-off-by	
Status Mod., not staged	File eclipse-swordfish: org.eclipse.swordfish.api/src/main/java/org/eclipse
()) () ()
\subset	Select All Deselect All Cancel Commit

We press "Commit", afterwards we right-click on the root again and select "Team / Show in Resource History":



Ok, the change is done locally. Now our remote master has to be updated. In Git the command to do this is "PUSH". So we select "Push to …" from the Team context menu (Note that on the following screenshots I am using a clone of a local Git repository – if you push to Olivers repo, you'll certainly see a real remote URI):

000	Push To Another Repositories
Destination Git Re Enter the location of	pository the destination repository.
Configured remo origin: /Users/jki Custom URI:	te repository: ndler/Documents/workspace_swordfish_git2//workspace_swordfish_git/eclipse-swordfish/ 🛟
URI: Host: Repository path:	
Connection Protocol:	\$
Authentication User: Password:	
0	< Back Next > Finish Cancel

Now we'll have fun with this dialog. For a first-timer it's hard to get.

0 0

Push To: origin

GIT

Push F	Ref Spe	cificat	tions
--------	---------	---------	-------

Select refs to push.

		Destination ref:	•	🛉 Add spec
dd delete ref	specificaton			
emote ref to c	lelete:		•	🔀 Add spec
dd predefined	specification			
Add co	nfigured push specs	Add all branches spec	Add a	ll tags spec
	inguice pass spees			in tugo spec
pecifications f	or push			
Mode	Source Ref	Destination Ref	Force Update	Remove
	·	·		
		Force update a	all specs 🕞 R	emove all specs

Currently we are on the local master ("master [branch]"), so we simply select this as source. The remote master will automatically selected and we press "Add spec":

0 0		Push To: origin		
Push Ref Specific	ations			GIT
Select refs to push.				
Add create/upd:	te specificaton			
Source ref:	the specification	Destination ref:		
*	•	*		Add spec
Add delete ref s	pecificaton			
Remote ref to de	lete:			🔀 Add spec
Add predefined	specification			
Add con	figured push specs	Add all branches spec	c Add a	III tags spec
Specifications fo	r push			
Mode	Source Ref	Destination Ref	Force Update	Remove
🐈 Update	refs/heads/master	refs/heads/master		Û
		Force update	all specs 🕞 🕞	emove all specs
0		< Back Nex	xt > Finish	Cancel
				11.

Note that we should not check the "Force Update" when pushing as it would overwrite the remote origin with our local one. It may be tempting if there is a conflict, but forcing is only OK for incoming changes (to overwrite local changes that do not exist in the remote origin).

So when we press "Finish" we get this:

ode	Source Ref	Destination Ref	Status: Repo #1
+	refs/heads/master	refs/heads/master	914b7b6bd6f255

Congratulations – the remote repository has been updated! Now the CI build based on Olivers Git will be triggered. In the mean time, we can create a patch file that includes our fix (and attach it to a

Bugzilla request. To do that we need a terminal window and use the command line tool. We enter the eclipse-swordfish directory of our workspace and enter

git log

0	0			Terminal	— less — 122	×16				
8	java	8	bash	\otimes	less	8	bash	8	bash	
commit Author: Date:	bd6f25575d0b7aa25 JuergenKindler < Wed Apr 1 14:44:	bf4c42632 juergen.k 44 2009 +	6c78c952cf265c indler@sopera.d 0200	e>						Ì
Ado	led a GIT testcomm	ent								
commit	commit 914b7b6b62b9eff4d5ee0d98b395f476cb56fce6									
Author: Date:	Author: renat.zubairov <renat.zubairov@sopera.de> Date: Tue Mar 31 15:42:11 2009 +0200</renat.zubairov@sopera.de>									
GIT	HOWO Test									
commit Author: Date: :	97ef1962cad2bea18 renat.zubairov < Tue Mar 31 15:38	b05a691b6 renat.zub :53 2009 -	bf59fee7cb8b1b airov@sopera.de +0200	>						

Now we copy the ID of the previous commit and create the patch file using

git format-patch 914b7b6b62b9eff4d5ee0d98b395f476cb56fce6

00	● ● ● Terminal — bash — 121×9									
8	java	8	bash	8	bash	8	bash	8	bash	
jurgen 0001–A jurgen	⊨kindlers-macbo dded-a-GIT-test ⊢kindlers-macbo	ok-pro:ec comment.po ok-pro:ec	lipse-swordfi atch lipse-swordfi	sh jkindler sh jkindler	\$ git format	-patch 914b7	'b6b62b9eff4d5	ee0d98b395f476c	b56fce6	

Git has created a patch for the commit that happened since this previous one. Let's have a look at what is inside:

00	0			Termi	nal — less -	– 135×27				
8	java	8	bash	\odot	less	8	bash	8	bash	
From b From: Date: Subjec 1 fil	d6f25575d0b7aa25b JuergenKindler <j Wed, 1 Apr 2009 1 t: [PATCH] Added ¤rg/eclipse/swordf es changed, 1 ins</j 	f4c426326c7 uergen.kind 4:44:44 +02 a GIT testc ish/api/Fil ertions(+),	%c952cf265c Mon Se; Ner@sopera.de> ∞mment terStrategy.java , 1 deletions(-)) 17 00:0	0:00 2001					
diff - /java/ index a/ +++ b/ @@ -18 * Tr * ba * Su - * + * !0 */	<pre>1 Tries changed, 1 insertions(+), 1 deletions(-) diffgit a/org.eclipse.swordfish.api/src/main/java/org/eclipse/swordfish/api/FilterStrategy.java b/org.eclipse.swordfish.api/src/main /java/org/eclipse/swordfish/api/FilterStrategy.java index 8f97581919d00f 100644 a/org.eclipse.swordfish.api/src/main/java/org/eclipse/swordfish/api/FilterStrategy.java +++ b/org.eclipse.swordfish.api/src/main/java/org/eclipse/swordfish/api/FilterStrategy.java @@ -18,7 +18,7 @@ import java.util.List; * The strategy that can remove some interceptors from the interceptor chain * based on the supplied hints. * Supposed to be plugged into the Swordfish framework as an osgi service * + * IGIT is GIT!</pre>								/main	
publi 1.6.1.	c interface Filte 3	rStrategy e	extends Strategy {							0
(END)										Y

Note that you can go backwards to previous patches as well. Each commit will end up in a separate patch file.

Committing a fix that causes conflicts

OK, the previous change was pretty simple ... pretty unrealistic also ;-). Let's have a look what happens if somebody else created a local clone of Olivers repository before we pushed our change to there. So the change is done and committed to the local Git (note that origin/master is one step behind us):



OK, so we want to be documentation heroes and push this change to the origin repository:

ode	Source Ref	Destination Ref	Status: Repo #1
÷	refs/heads/master	refs/heads/master	(rejected)

So somebody was faster - our change is rejected because of a merge conflict. :-(

We need to know what happened on the origin repository and fetch its contents:

0 0	Fetch From: origin		
Fetch Ref Specifications Select refs to fetch.			GIT
Add create/update specificaton Source ref: * Add predefined specification Add configured fetch spece	Destination ref:	spec Add	Add spec
Specifications for fetch Source Ref refs/heads/master	Destination Ref refs/remotes/origin/master Force up	Force Update	Remove
Annotated tags fetching strategy Automatically follow tags if we fe Always fetch tags, even if we do Never fetch tags, even if we have Save specifications in "origin" conf	atch the thing they point at not have the thing it points at the thing it points at iguration		
0	< Back	Next > Finish	Cancel

So we are able to get the changes:

urce Ref	Destination Ref	Status		
s/heads/master	refs/remotes/origin/master	914b7b6bd6f255		

We can also located them outside Eclipse using git diff origin:

\odot \bigcirc \bigcirc)			Terminal	- less - 131>	<17				
\otimes	java	\otimes	bash	\otimes	bash	8	git	\odot	less	
diffgi1 index 9190 a/org +++ b/org @@ -18,7 - * The st * based * Suppos +<<<<<< h colspan="2">* * * * * * * * * * * * * * * * * * *	c a/org.eclipse.s 100f76cde0d 100 .eclipse.swordfis .eclipse.swordfis .18,11 @@ import .rategy that can on the supplied sed to be plugged HEAD:org.eclipse. is GIT!	wordfish.a)644 xh.api/src/ yh.api/src/ java.util. remove som hints. I into the .swordfish.	pi/src/main/java main/java/org/ec List; e interceptors f Swordfish framew api/src/main/jav	/org/eclip lipse/swor lipse/swor rom the in ork as an a/org/ecli	ose/swordfish/api dfish/api/Filter dfish/api/Filter derceptor chain osgi service. pse/swordfish/ap	/FilterStr Strategy.j Strategy.j Di/FilterSt	ategy.java b/orq ava ava rategy.java	g.eclipse.s	wordfish.api/src	1
+ * GIT is +>>>>>> / */ public ir (END)	s NOT SVN! A conflicting cho hterface FilterSt	nge!:org.e	clipse.swordfish ends Strategy {	.api∕src/m	ain/java/org/ec	ipse∕sword	fish/api/FilterS	∂trategy.ja	va	

But now we have to leave Eclipse and open a Terminal window to rebase with the origin repository, because that is not yet supported in the Egit plugin. We enter git rebase origin and Git tries to merge changes. In our case it fails, because of the conflicting line:



Now contents in our Eclipse editor change:



We resolve the conflict by keeping both changes and save our file:



That done we do not have to commit, but have to (re-) add¹ the changed file to indicate that we have solved the conflict using

```
git add
org.eclipse.swordfish.api/src/main/java/org/eclipse/swordfish/api/FilterStrategy
.java
```

Now the rebase can continue:

git rebase -continue

● ○ ○ Terminal — bash — 131×17										
8	java	8	bash	8	bash	8	git	8	bash	
jkindler:eclipse-swordfish jkindler\$ git add org.eclipse.swordfish.api/src/main/java/org/eclipse/swordfish/api/FilterStrategy.java 📗										
jkindle	er:eclipse-swordfish	n jkindler\$	git rebasec	ontinue						
Applyin	Applying: A conflicting change!									
ikindle	er:eclipse_swordfisk	h ikindler≸								

1 Curious who implemented git add? Then type git add -help and go to the bottom of the help text ;-)

• •

And having done that, our push will work:

$\Theta \cap O$		Push Results: origin	Push Results: origin					
Pushed to o	origin.							
Mode	Source Ref	Destination Ref	Status: Repo #1					
+	refs/heads/master	refs/heads/master	bd6f2556bc2970					
			ОК					

Again we can see that on the terminal window as well using git log:

0	0			Terminal	— less — 133×	:17				
8	java	\otimes	bash	8	bash	8	bash	0	less	
commit Author: Date:	6bc29702780e27a50f5 : JuergenKindler <ju Wed Apr 1 15:46:55</ju 	470a352e58 ergen.kind 2009 +020	38a8f5cfa930 1ler@sopera.de> 30							Ô
A c	conflicting change!									L
commit bd6f25575d0b7aa25bf4c426326c78c952cf265c Author: JuergenKindler <juergen.kindler@sopera.de> Date: Wed Apr 1 14:44:44 2009 +0200</juergen.kindler@sopera.de>							l			
Add	led a GIT testcommen	t								L
commit Author: Date:	914b7b6b62b9eff4d5e : renat.zubairov ⊲re Tue Mar 31 15:42:1	e0d98b3951 nat.zubai1 1 2009 +02	f476cb56fce6 rov@sopera.de> 200							0
:[]										▼ //.

Resetting the local repository clone to the state of its origin

Suppose we have made some changes that we do not want to push to the origin repository. In that case we need to go back to the last state of the origin (as we can see below, our local master branch is 3 commits ahead of the last known state of the origin master branch).



With EGit you will have to get the last changes of the origin and then reset you local branch to the state of the origin. So right-click on the eclipse-swordfish project and select "Team / Fetch from ..."

00	Fetch From Another Repository	
Source Git Repository Enter the location of the source re	pository.	GIT
Configured remote repository: Origin: /Users/jkindler/Docume Custom URI: Location URI: Host: Repository path: Connection Protocol: Port: Authentication User:	ents/workspace_swordfish_git2//workspace_swordfish_git/eclipse-swor	dfish/
0	< Back Next > Finish	Cancel

Press "Next >" ...

0 0	Fetch From: origin	ı	
Fetch Ref Specifications Select refs to fetch.			GIT
Add create/update specificaton			
Source ref:	Destination ref:	•	Add spec
Add predefined specification			
Add configured fetch spe	cs Add all branche	s spec Add	all tags spec
Source Ref	Destination Ref	Force Update	Remove
refs/heads/*	refs/remotes/origin/*		Û
Annotated tags fetching strategy Automatically follow tags if w Always fetch tags, even if we Never fetch tags, even if we h 	Force update re fetch the thing they point at do not have the thing it points at nave the thing it points at	all specs	temove all specs
Save specifications in "origin" o	configuration < Back Ne	ext > Finish	Cancel
			1.

Then "Finish" it:

ource Ref	Destination Ref	Status

So there were no changes on the origin since our last fetch command.

Now we can reset our branch to the contents of the origin. We use "Team / Reset To ...", select the master branch of the origin repository we just fetched and select the Reset Type "Hard" (overwrite everything locally!):

000	
HEAD Local Branches master (curr Remote Branches origin master	ent)
	Reset Type Soft (Index and working directory unmodified) Mixed (working directory unmodified) Hard Cancel

No we will have to confirm that we really want to wipe out all local changes (the answer is Yes!)

00	Really reset?
1	Resetting will overwrite any changes in your working directory. Do you wish to continue?
	No Yes

So when the operation is finished, we see that our local master branch and the remote master branch are at the same state:

\varTheta 🔿 🔿 Java – org.eclipse.swordfish.api/src/main/java/org/eclipse/swordfish/api/FilterStrategy.java – Eclipse SDK – /Users/jkindler/D 😑					
📬 • 🔚 📄 🏇 • 💽 • 🂁 😫 😫	방 @ •] 🧐 😂 😂 🖋 •] 약 🥖 💝] 삼 • 상 - * 수 • 수 • 수 • 한 🗄 🔡 Java				
📲 Package Expl 🖾 🍃 Hierarchy 🗖 🗖	🛛 FilterStrategy.java 🕱				
 Filtrage Expl 23 Filter Strategy.java Filtrage Filtrage Filtrage Filtrage Filtrage Filtrage Filtrage Filtrage Filtrage Filtrage Filtrage Filtrage Filtrage	<pre> PrinceStrategy java as PrinceStrategy java as PrinceStrategy java as PrinceStrategy is a constructed in the supplied hints. * Supposed to be plugged into the Swordfish framework as an osgi service. * IGIT is GIT! * GIT is NOT SVN! */ public interface FilterStrategy extends Strategy { PrinceStrategy is a constructed in the interceptors which * shall process subsequent message exchanges. The following rules apply * for filtering: * </pre>				
 Registry.java SortingStrategy.java Strategy.java SwordfishException.jav org.eclipse.swordfish.api.c 	 The original interceptor chain must not be modified. Either the unmodified List of interceptors is returned, or a new List with the interceptory № History № Image: A state of the filter is and the filter is an an	<. < □ □			
 org.eclipse.swordfish.api.c org.eclipse.swordfish.api.e 	Author	Date			
org.eclipse.swordfish.api.p	master origin/mast [HEAD] A conflicting change! JuergenKindler <juergen.kindler@: 2<="" th=""><th>2009-04-0</th></juergen.kindler@:>	2009-04-0			
org.eclipse.swordfish.api.re	Added a GIT testcomment JuergenKindler < juergen.kindler@: 2	2009-04-0			
Figure 2015 JRE System Library [JVM 1.5.0 (M	GIT HOWO Test renat.zubairov < renat.zubairov@s 2	2009-03-3 🟹			
Referenced Libraries	Modification to test GIT howto renat.zubairov <renat.zubairov@s 2<="" th=""><th>2009-03-3 🔻</th></renat.zubairov@s>	2009-03-3 🔻			
🕨 🧁 META-INF					
🕨 🧁 src	commit 6bc29702780e27a50f5470a352e588a8f5cfa930				
target	15.46.55 M org.eclipse.sword	dfish.api/sr			
pom.xml	Committer: JuergenKindler <juergen.kindler@sopera.de></juergen.kindler@sopera.de>				
org.eclipse.swordfish.bundles	2009-04-01 16:31:29				
ord eclipse swordtish compatibility	Parent: bd6f25575d0b7aa25bf4c426326c78c952cf265c (Added a GIT				
		1			