Korišćenje CodeCover - Od koda do izveštaja

Uvod

Ovaj dokument opisuje korišćenje CodeCover Eclipse plugin. Koristićemo Java aplikaciju "SimpleJavaApp". Ova aplikacija je sistem koji se testira ("system under test", SUT). U ovom dokumentu opisaćemo CodeCover instrumente i izvršavanje sistema koji testiramo, i kako rezultati testa mogu biti vizuelno prikazani i prikazani u vidu izveštaja.

Preduslovi za korišćenje tutorijala su:

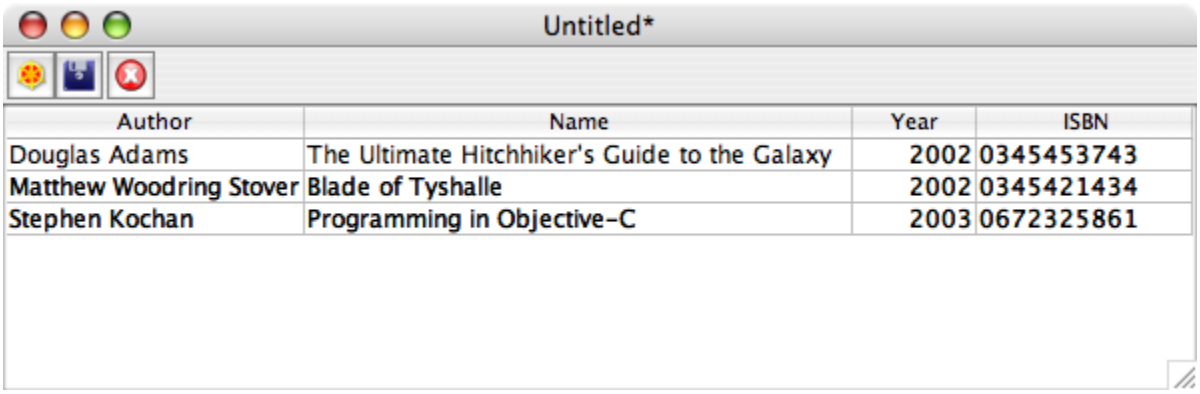
- Osnovno poznavanje Eclipse (učitavanje projekta, pokretanje projekta, otvaranje različitih prozora i pogleda,...).
- Instaliran CodeCover Eclipse plugin
- Podaci za ažuriranje Eclipse-a:

```
Name: CodeCover Update Site
URL:  http://update.codecover.org/
```

Program SimpleJavaApp

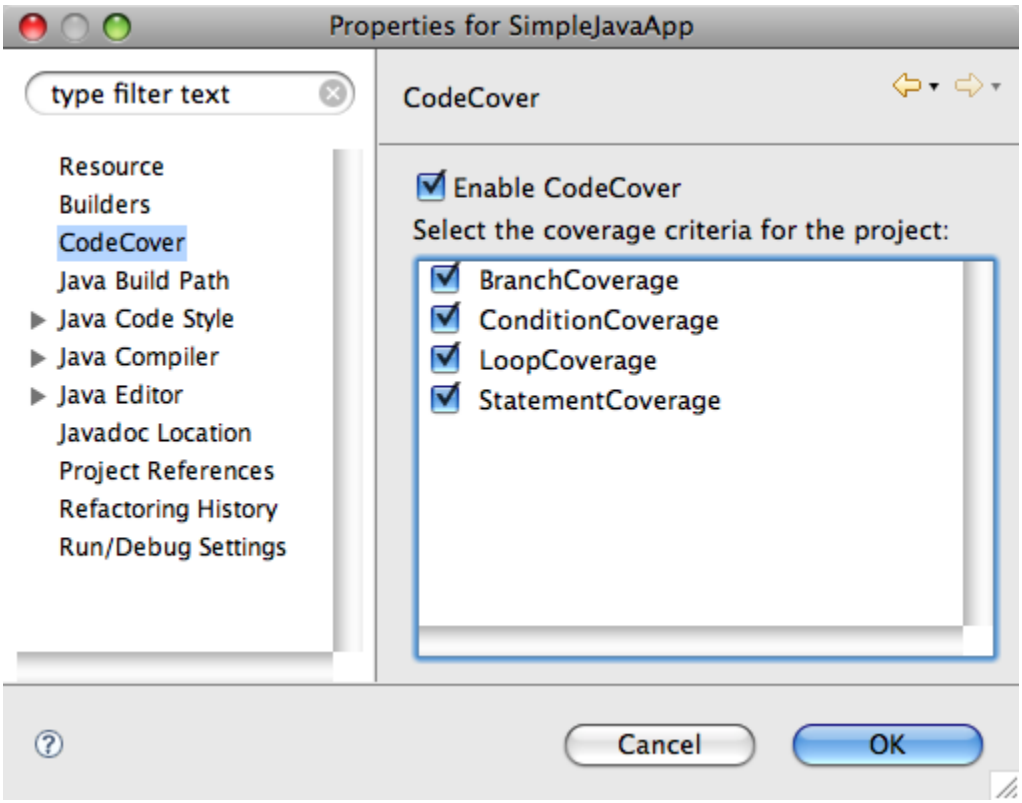
"SimpleJavaApp" je mala Swing aplikacija, koja se koristi za prikazivanje i editovanje liste knjiga. Možemo snimiti ili učitati listu knjiga pomoću XML fajlova, kao što je prikazano na slici ispod.

Otvorićemo Eclipse i importovati "SimpleJavaApp" u naš radni prostor. Odaberimo opciju "Import" u meniju "File" i odaberimo "Existing Projects into Workspace" u kategoriji "General". Pronađimo folder gde se nalazi aplikacija.



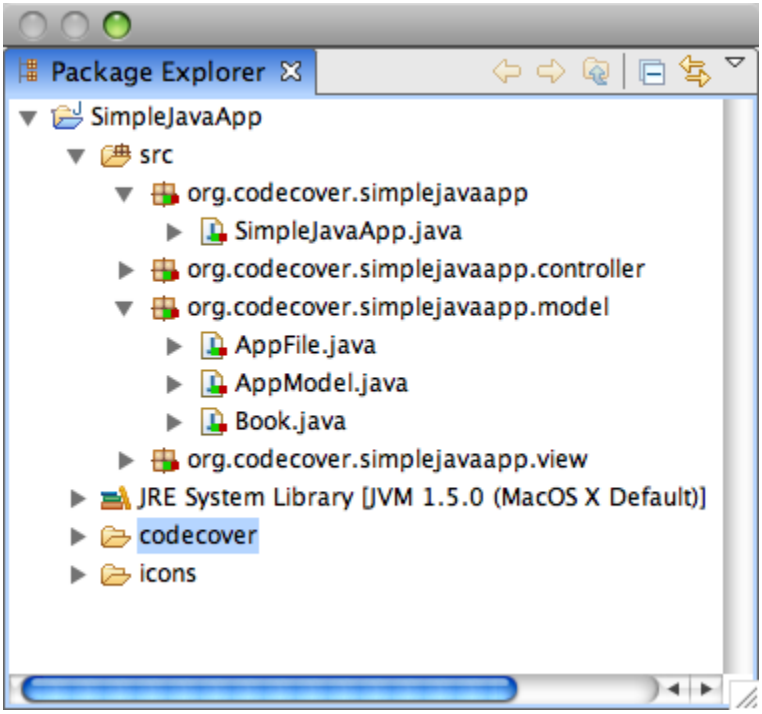
Dozvoliti korišćenje CodeCover

Potrebno je da dozvolimo korišćenje dodatka CodeCover za projekat koji smo importovali. Otvorićemo properties opciju na projektu i odabrati CodeCover kategoriju. Izaberimo checkbox kao što je prikazano na slici ispod. Sada smo aktivirali CodeCover za projekat "SimpleJavaApp". Takođe, potrebno je da odaberemo kriterijume za pokrivenosti koje će se koristiti. U našem slučaju odabrali smo sve dostupne kriterijume.



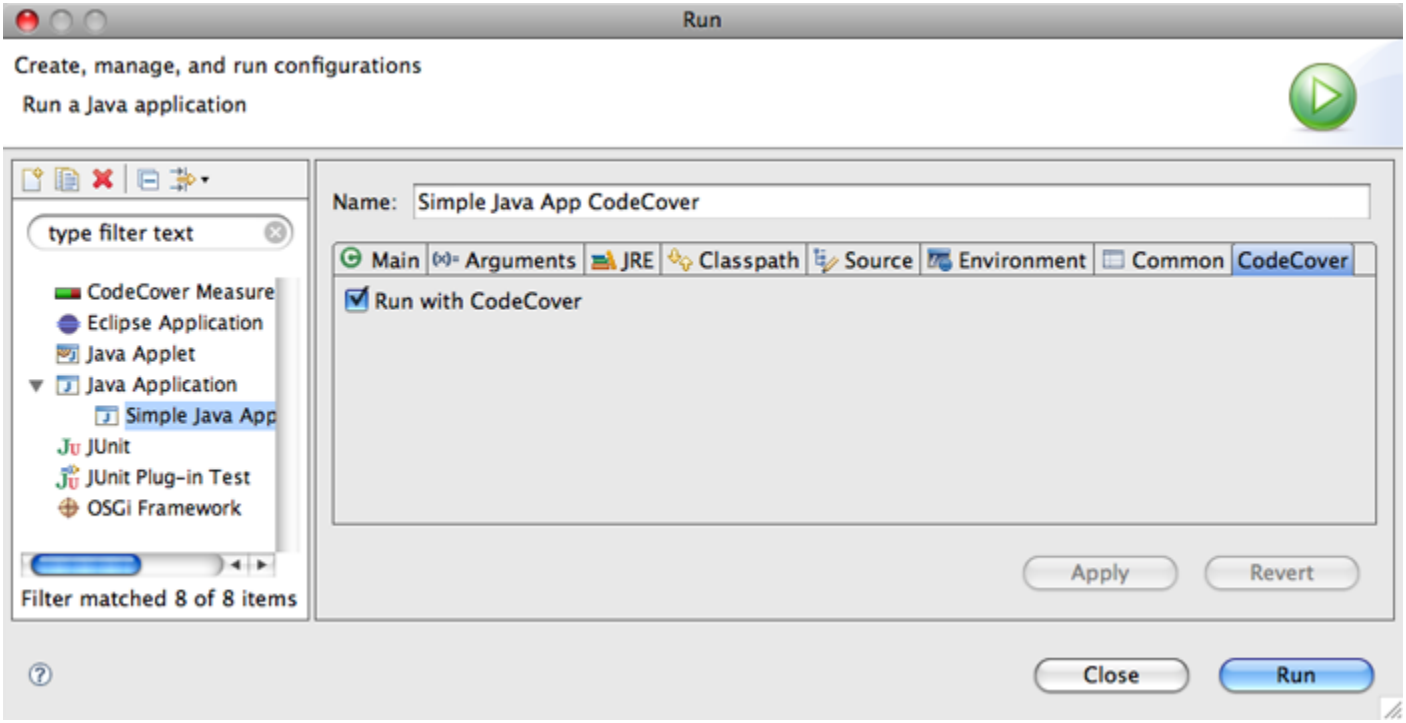
Kako se meri pokrivenost

Odaberimo klase koje želimo da posmatramo. Otvorićemo “package explorer view” i u izvornom direktorijumu projekta "SimpleJavaApp" izabrati klase koje želimo da posmatramo. Kliknimo desnim dugmetom miša i odaberimo "Use For Coverage Measurement" za svaku klasu koju želimo da označimo (istovremeno će se promeniti ikonica). U ovom primeru, sve klase projekta "SimpleJavaApp" su označene za praćenje pokrivenosti.



Pokretanje aplikacije sa CodeCover

Da bismo koristili CodeCover, potrebno je da pokrenemo aplikaciju u tom režimu, tako što u "run configuration" dijalogu naglasimo da koristimo taj dodatak. Potrebno je u CodeCover tab-u odabrati opciju “Run with CodeCover”, kao što je prikazano na slici.



Izvršavanje

Postoji nekoliko načina da se izvrši sistem koji se testira. Možemo izabrati pokretanje aplikacije kao i obično, i u tom slučaju svi izmereni podaci se prikupljaju u jedan test. Druga varijanta je da koristimo “Live Notification” funkciju i snimamo pojedinačne testove. Treća varijanta je da koristimo postojeće JUnit grupe testova da definišemo naše testove.

Obično izvršavanje

Potrebno je samo da pokrene postojeću konfiguraciju. CodeCover se izvršava u pozadini. Pokrenimo testove i nakon završetka rada sa aplikacijom, merenja će automatski biti sačuvana kao testovi u test sesiji nazvanoj “eclipserun”. Merenja možemo pratiti u prozoru u Eclipsu ili kreiranjem HTML izveštaja iz tih podataka.

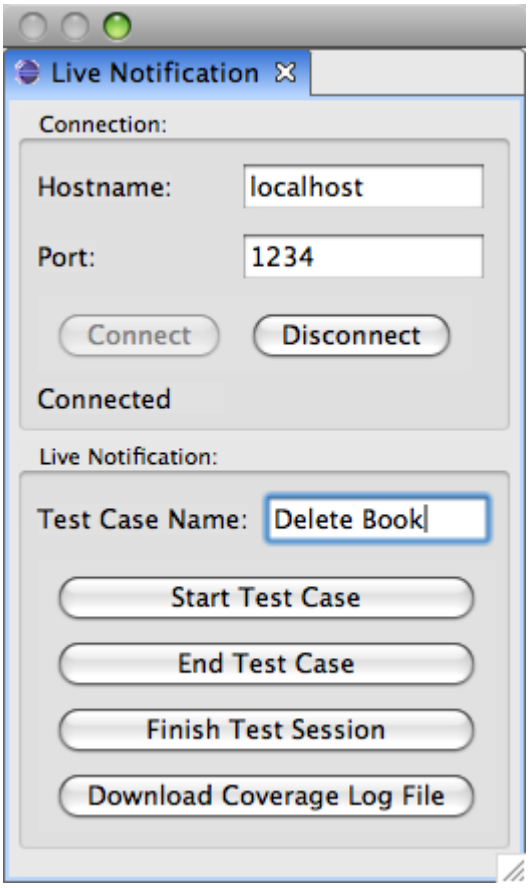
Live Notification izvršavanje

Da bismo koristili ovu opciju, potreban je još jedan dodatni korak, a to je da u tab sa argumentima konfiguracije za pokretanje programa dodamo sledeće parametre kao VM argumente:

```
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=1234
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
```

Pokrenimo aplikaciju sa ovom konfiguracijom, otvorićemo “Live Notification” prozor u Eclipse. Upisaćemo “localhost” za hostname i “1234” za port. Kliknimo na “Connect” dugme i “Live notification” će sada biti aktivan.

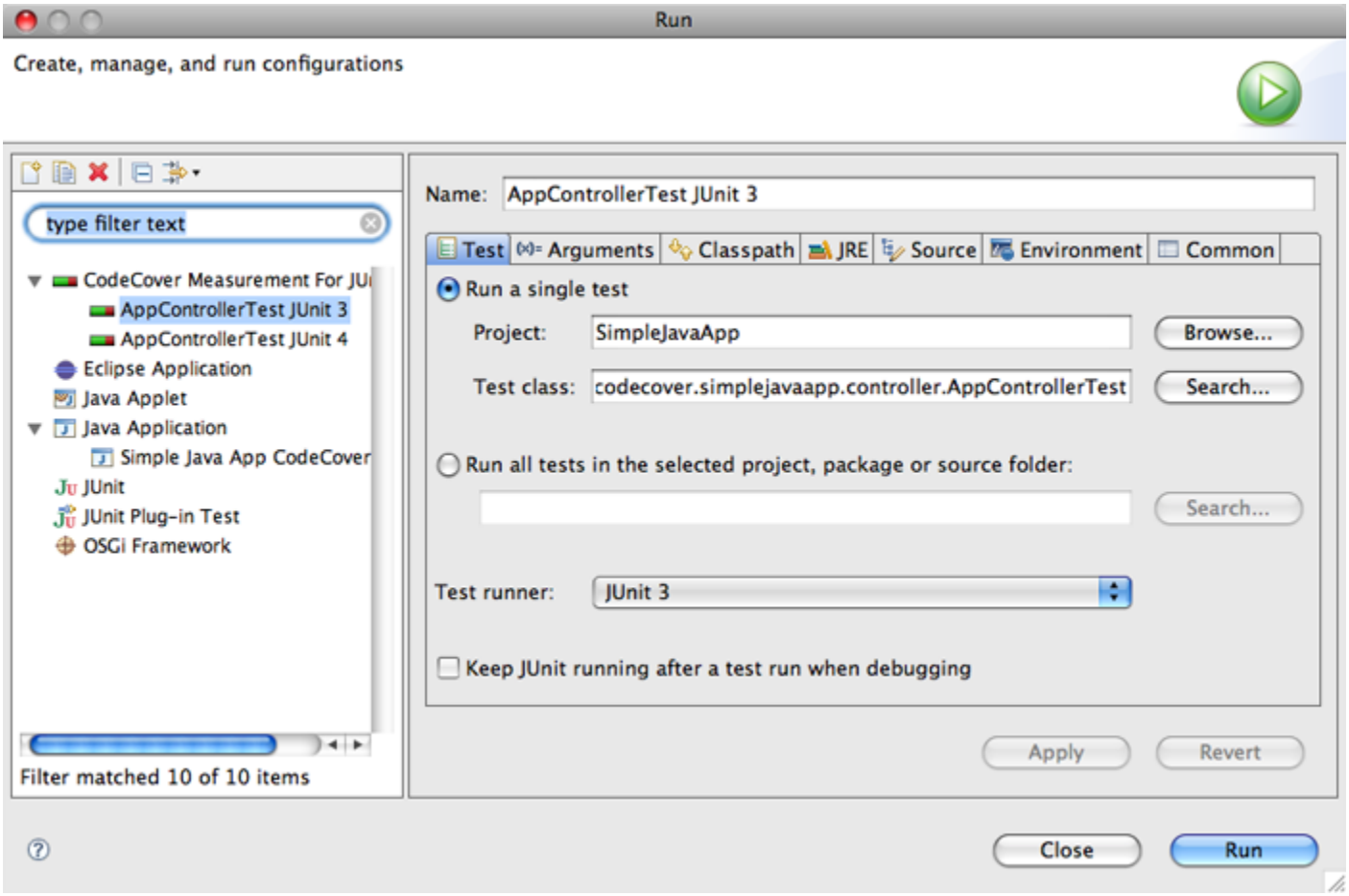
Upišimo ime u test i krenućemo da ga izvršavamo nakon što kliknemo na dugme "Start Test Case". Na primer izvršićemo brisanje knjige. Na kraju testa, pritisnimo dugme "End Test Case". Možete snimiti i više testova ponavljanjem ovih koraka. Konačno, da završimo snimanje testova, izabraćemo opciju "Finish Test Session". Ne moramo da koristimo opciju "Download Coverage Log File", zato što SimpleJavaApp nije Web aplikacija. Nakon izvršavanja SUT, merenja će automatski biti sačuvana.


Live Notification
Connection:
Hostname:
localhost
Port:
1234
Connect
Disconnect
Connected
Live Notification:
Test Case Name:
Delete Book
Start Test Case
End Test Case
Finish Test Session
Download Coverage Log File

JUnit izvršavanje

Da bismo koristili postojeće grupe testova, potrebno je da kreiramo novu konfiguraciju "CodeCover Measurement For JUnit". Kao što je prikazano na slici ispod, neophodno je da izaberemo klasu koja sadrži naše JUnit test slučajeve ili test suite.

Za izvršavanje testova možemo izabrati JUnit 3 ili JUnit 4 (opcija: test runner), zavisno od toga na čemu se zasniva naša postojeća grupa testova. Nakon toga ćemo pokrenuti aplikaciju koju testiramo, da bismo pratili merenja. Nakon izvršavanja, merenja će automatski biti zabeležena u test sesiji nazvanoj “eclipserun”, koja čuva sve testove koji su definisani u našoj grupi testova.

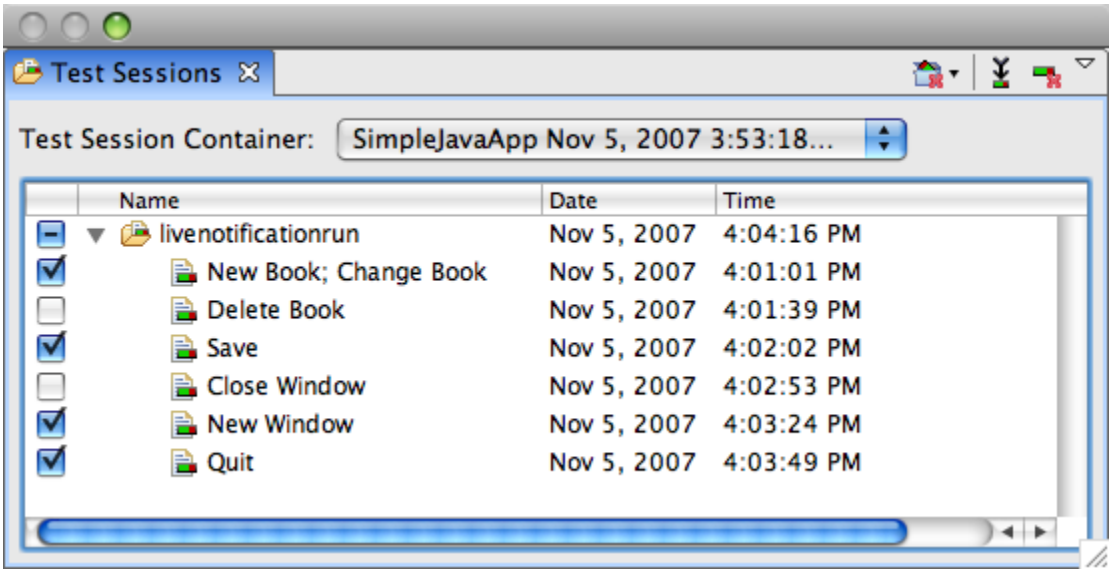


Pregled izmerenih podataka u Eclipse-u

U ovoj sekciji su opisane osnovne odlike koje CodeCover pruža.

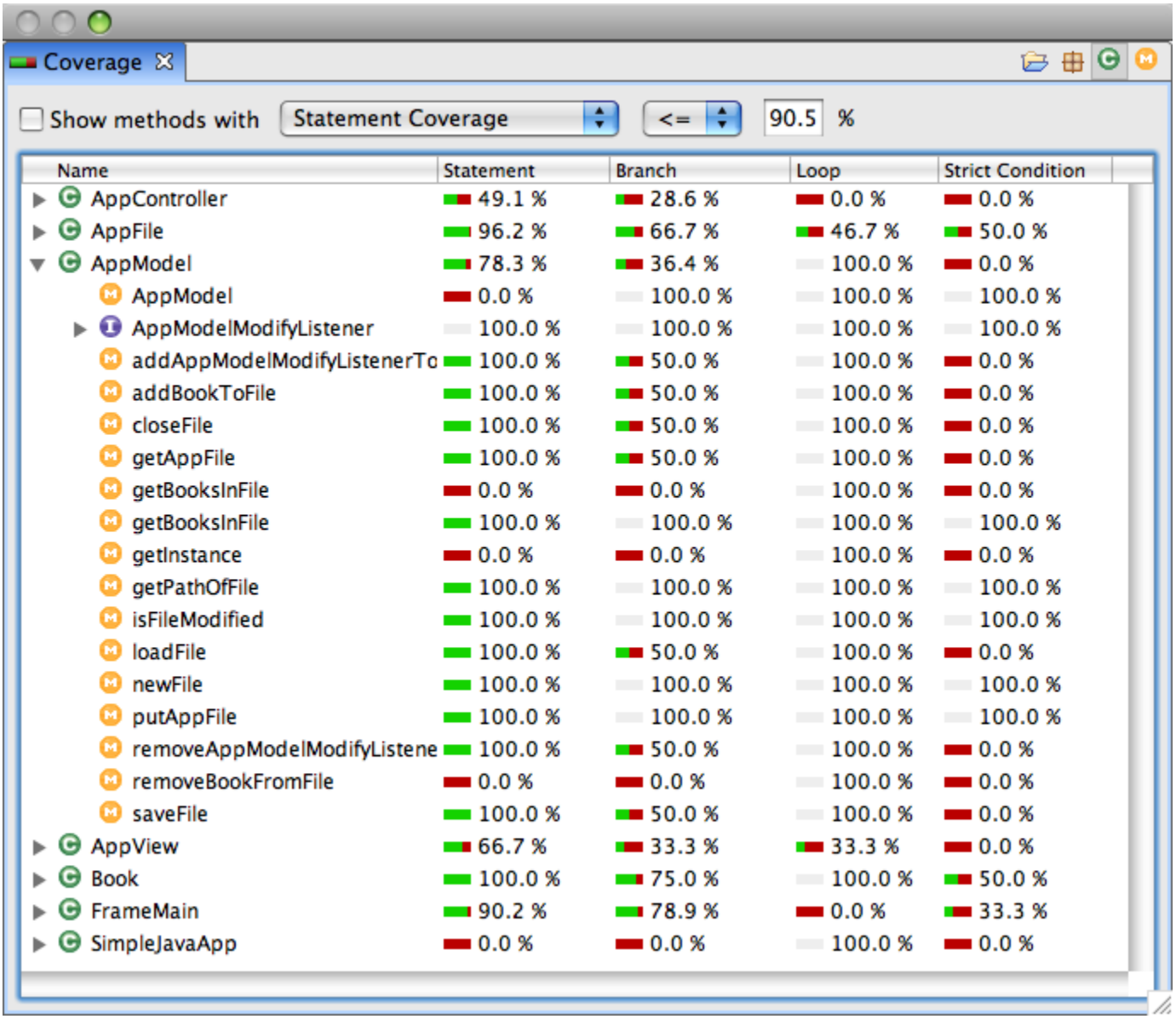
Test Sessions View

Ovaj pogled prikazuje test sesije i testove u grupama. Podržane su sledeće opcije: select, deselect, rename, delete i merge. Drugi pogledi koriste samo test slučajeve u prikazivanju.



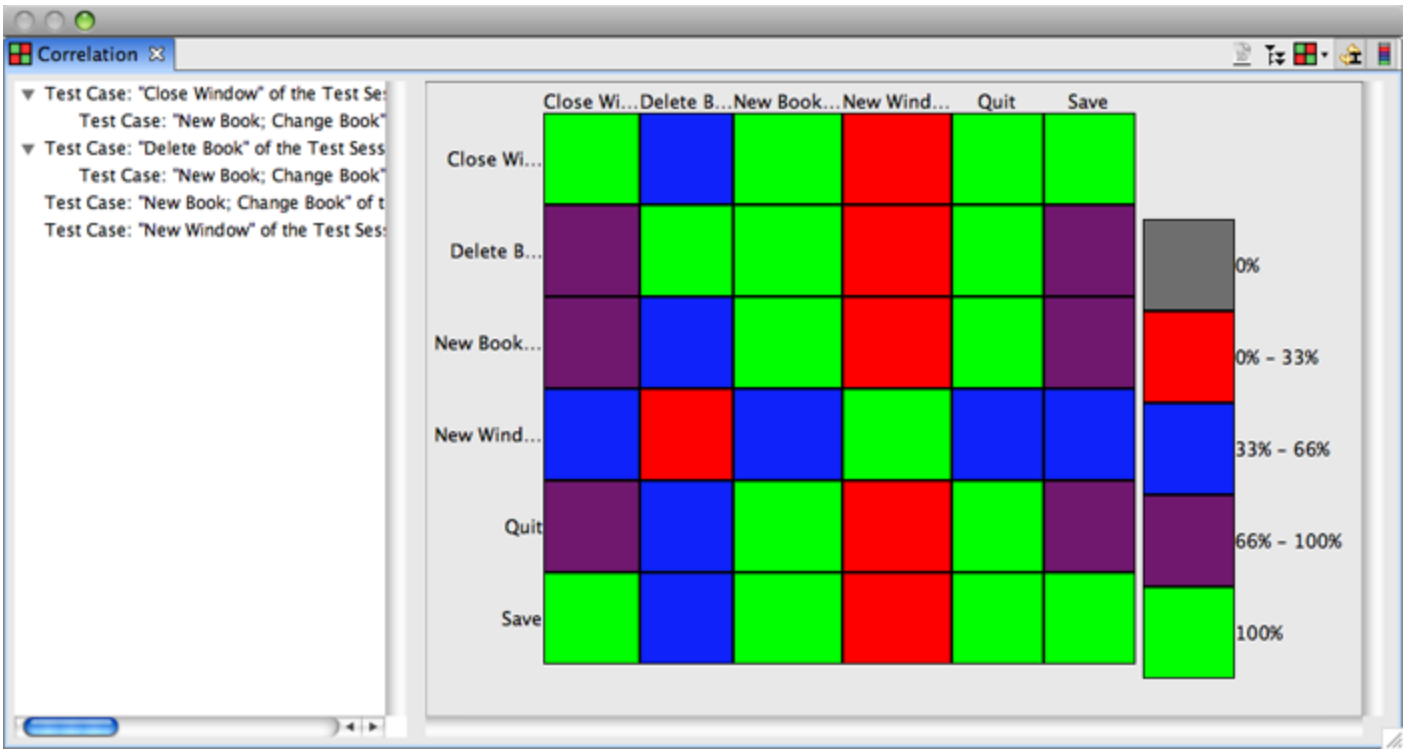
Coverage View

U ovom pogledu možemo videti pokrivenost određenih delova aplikacije koju smo testirali. Kao što je prikazano na slici ispod, svaka metrika ima svoju posebnu kolonu.



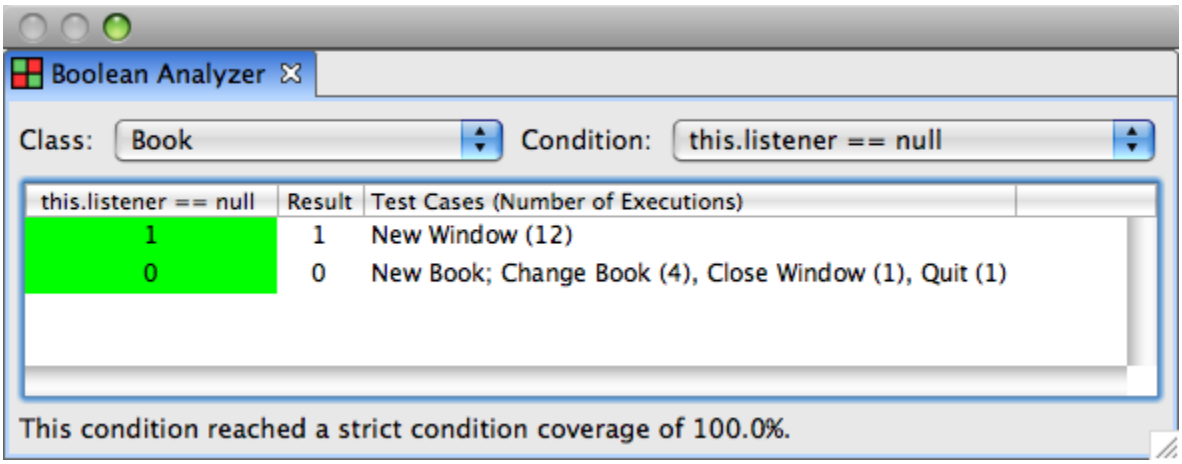
Correlation View

Ovaj pogled se koristi da uporedi test slučajeve sa ostalima. Možemo postaviti kursor miša iznad nekog bloka u matrici i videti koliko jedan test pokriva iste delove koda u odnosu na drugi test. Stablo prikazuje test koji u potpunosti pokrivaju delove koda koji drugi testovi pokrivaju, u hijerarhiji. Takođe, možemo eksportovati podatke iz matrice u fajl, koji je kompatibilan sa mnogim aplikacijama (Excel i druge).



Boolean Analyzer

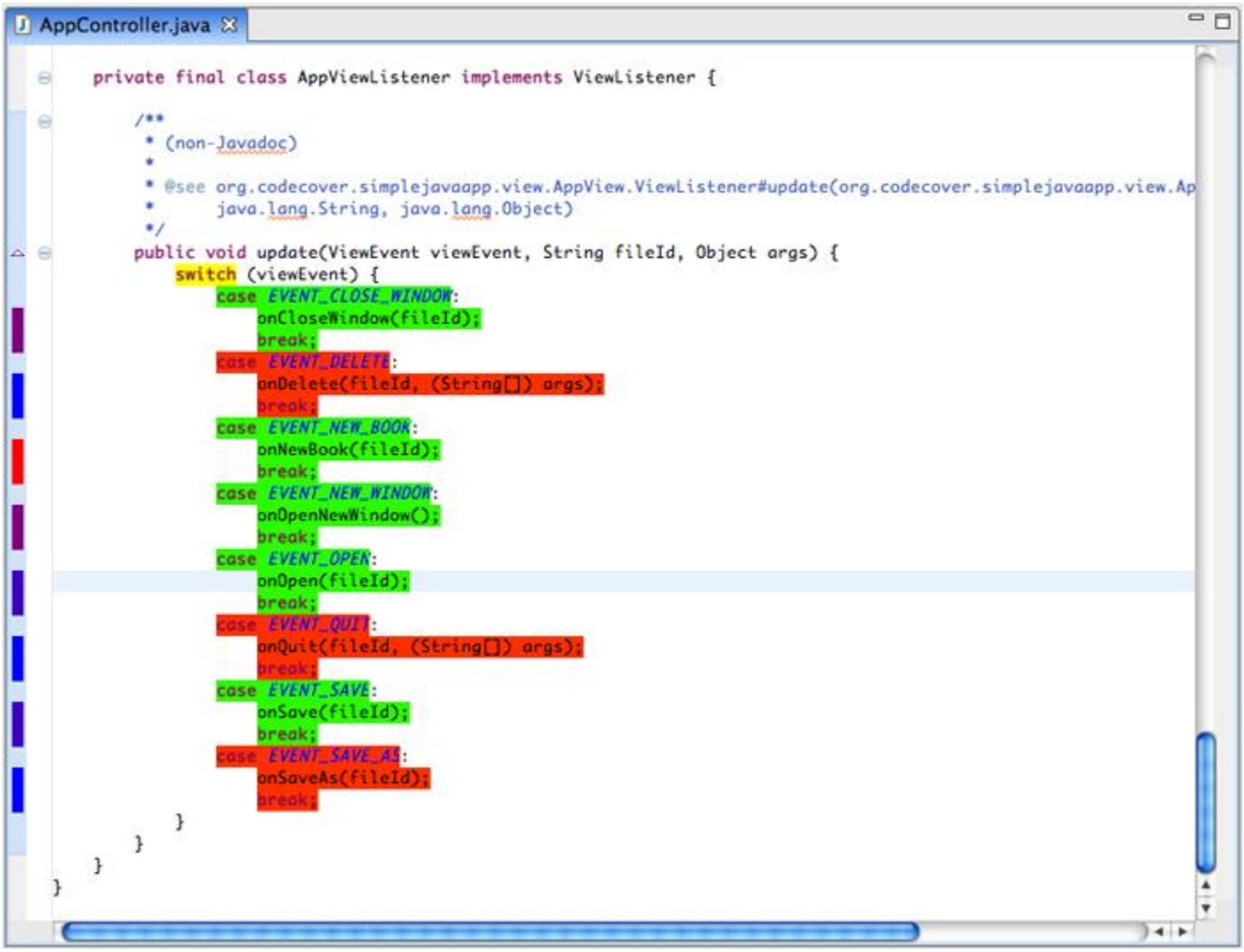
Ovim pogledom možemo videti zadati uslov i videti pokrivenost stanja. Potrebno je odabrati ključnu reč uslova u kodu, kliknuti desnim dugmetom miša i izabrati opciju “Analyze Term” u meniju, koja će automatski postaviti taj uslov u Boolean analizator.



Code Highlighting & Hot Path

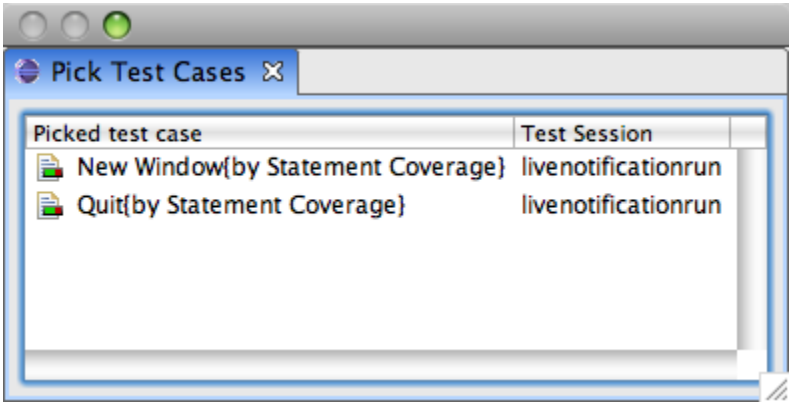
Izvorni kod može da se oboji na osnovu pokrivenosti. Taj prikaz je dat samo ukoliko izvorni kod nije menjan od poslednjeg merenja pokrivenosti. Kao što je prikazano na slici ispod, metode `onQuit(...)` i `onSaveAs(...)` se ne izvršavaju u vreme merenja pokrivenosti.

Putanja izvornog fajla koja se više izvršava je prikazan na levoj strani editora. Iskazi koji su češće izvršavani, postaju više crveni. Na primer metoda `onNewBook(...)` se izvršava mnogo češće nego druge metode, pa je crvene boje.



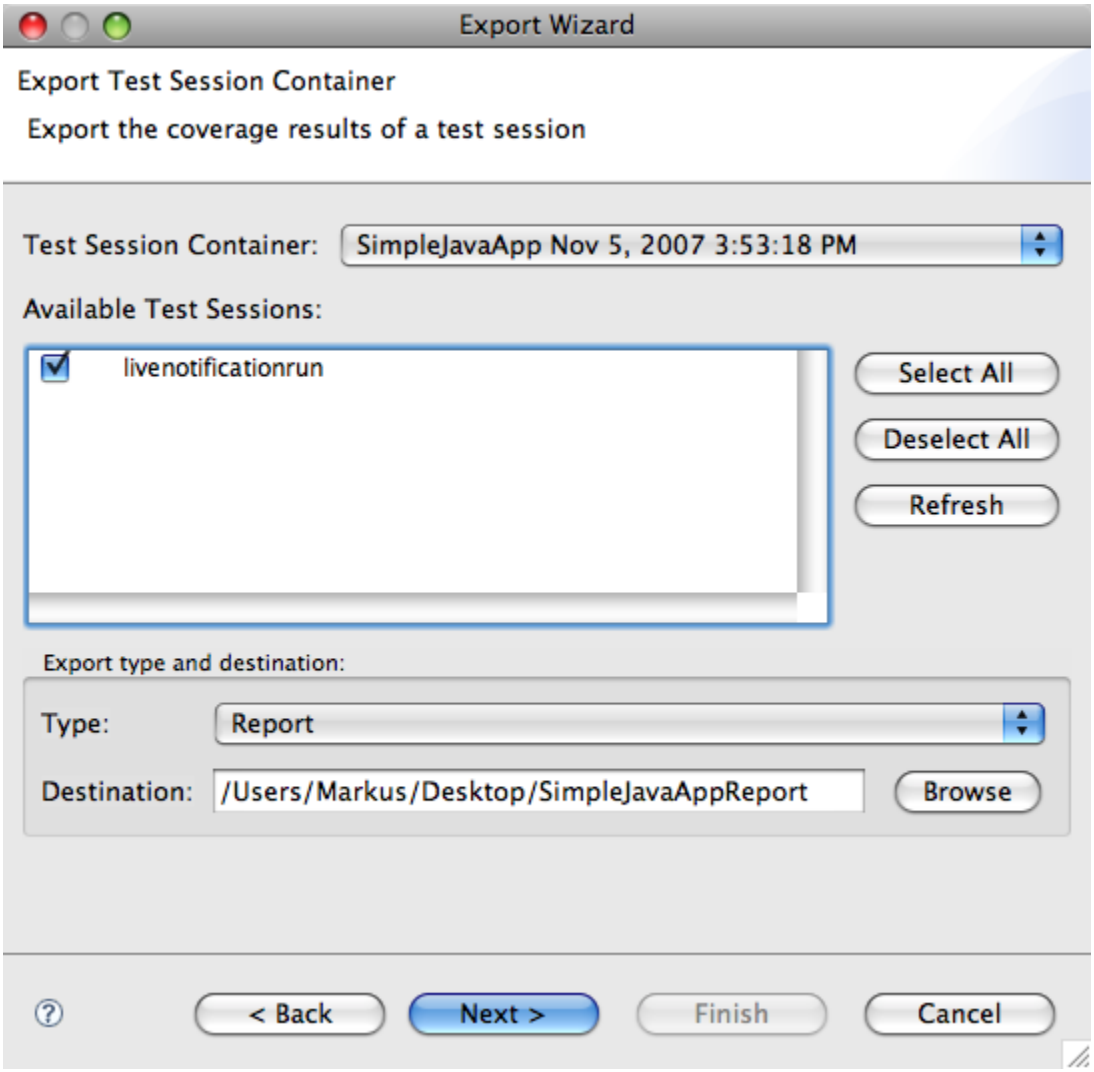
Pick Test Case View

Ovaj pogled prikazuje koji testovi pokrivaju koji deo koda, na primer koji testovi izvršavaju metod `onNewWindow()`. On prati ono što je selektovano u trenutno otvorenom editoru i prikazuje listu testova. Takođe se prikazuje i test sesija.

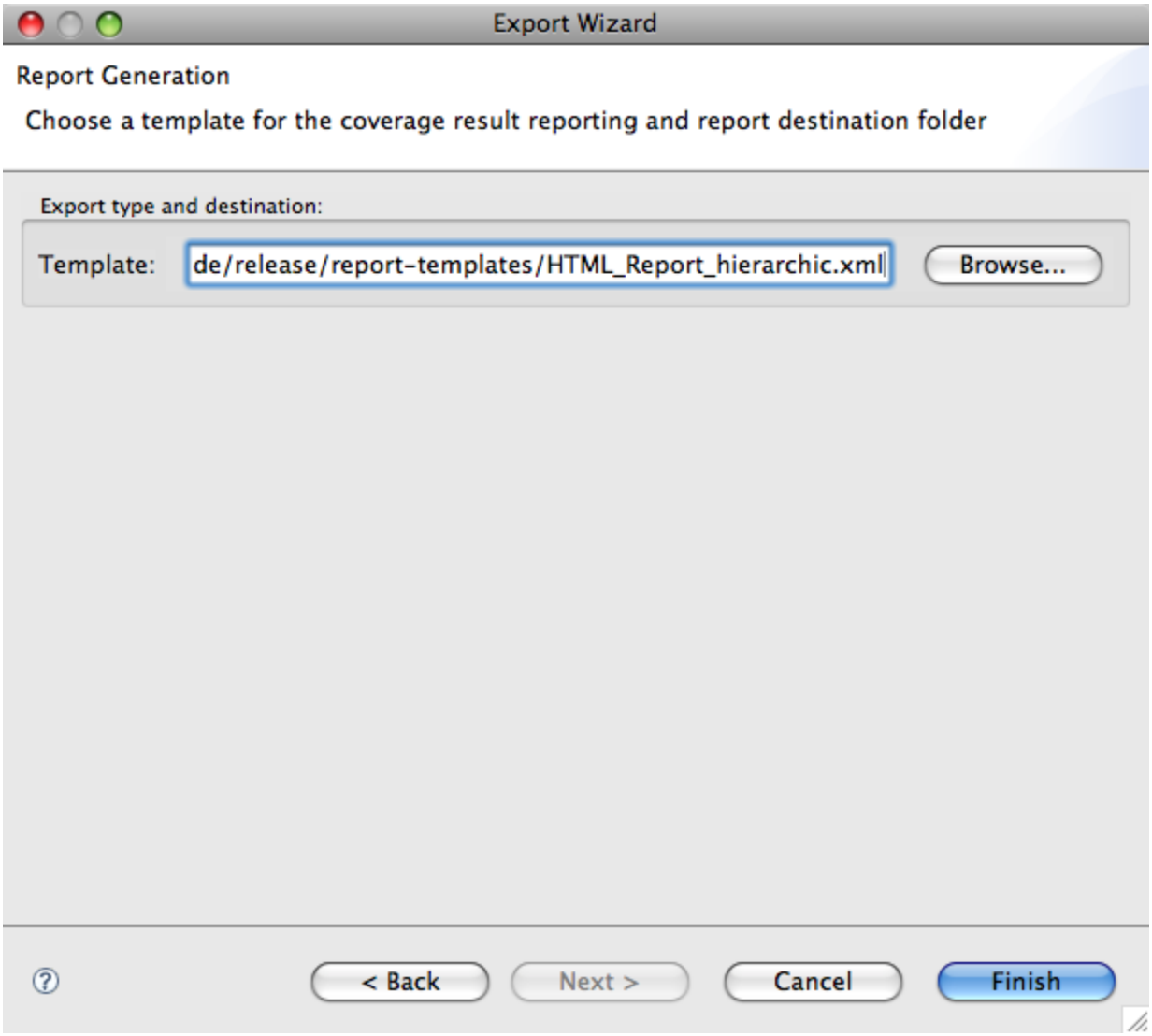


Export Report

Možemo eksportovati podatke kao HTML izveštaj. Odaberite opciju "Export" u "File" meniju u Eclipse-u i odaberite "Coverage Result Export" u kategoriji "CodeCover". Zatim izaberimo skup koji sadrži testove koje želimo da eksportujemo. Postavite tip za eksportovanje na "Report". Potrebno je da odaberemo i destinaciju za izveštaje, slično kao što je prikazano na slici.



Pritisnimo "Next" dugme da izaberemo šablon za izveštaj.



Izveštaj će biti kreiran na određenoj lokaciji, kada kliknemo na dugme “Finish”.