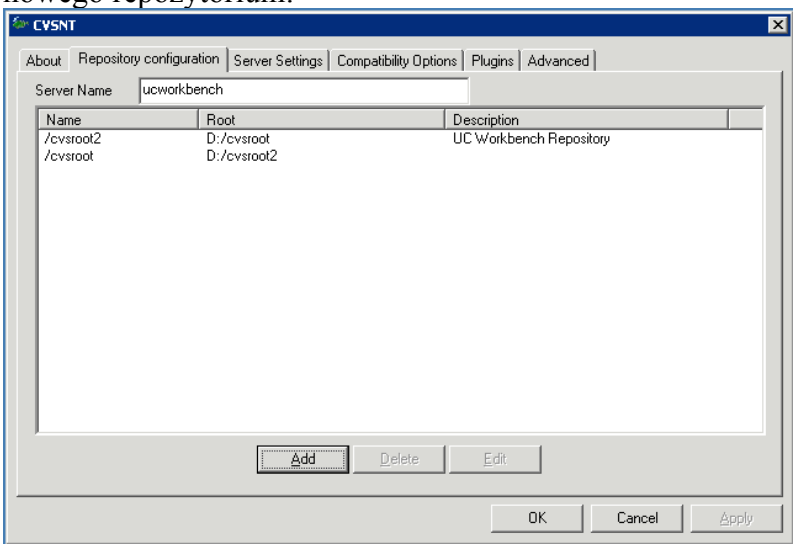
Ćwiczenia 9: Zarządzanie konfiguracją

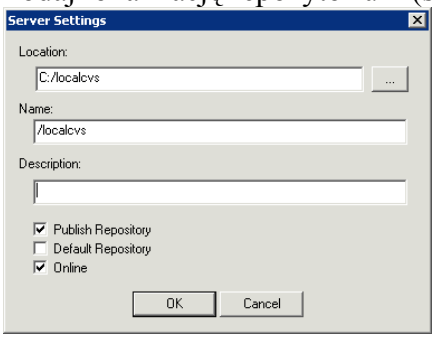
Zadania:

Konfiguracja repozytorium CVS:

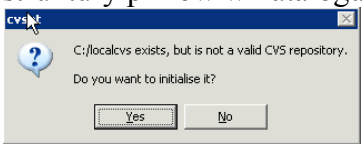
1. Ściągnij i zainstaluj serwer CVS: CVSNT (www.cvsnt.org).
2. W konfiguracji repozytoriów (Panel Sterowania -> CVSNT) wybierz dodanie nowego repozytorium:



3. Podaj lokalizację repozytorium (ścieżkę lokalną) oraz nazwę:

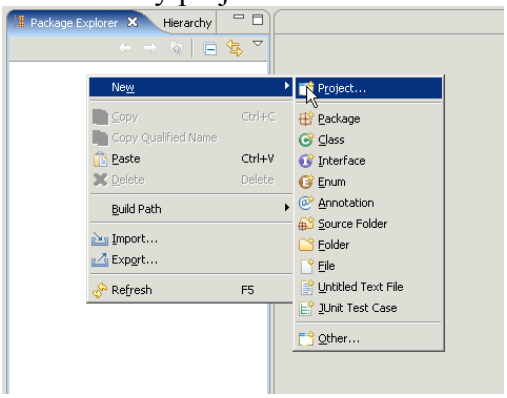


4. Poproś CVSNT o zinicjalizowanie repozytorium (utworzenie wymaganej struktury plików w katalogu)

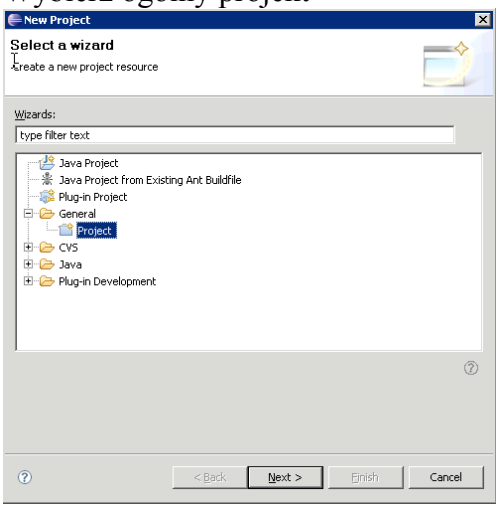


Konfiguracja środowiska Eclipse (które służy jako klient CVS) i utworzenie nowego projektu

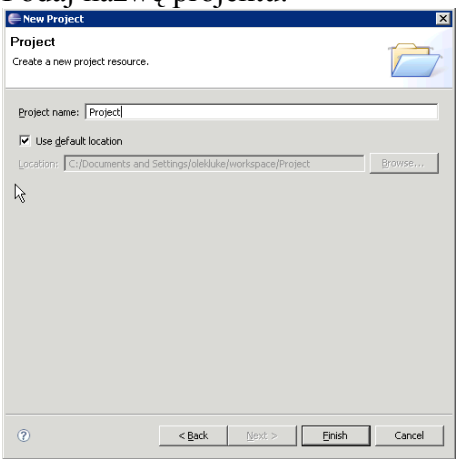
5. Pobierz i zainstaluj środowisko Eclipse (www.eclipse.org). Po uruchomieniu utwórz nowy projekt:



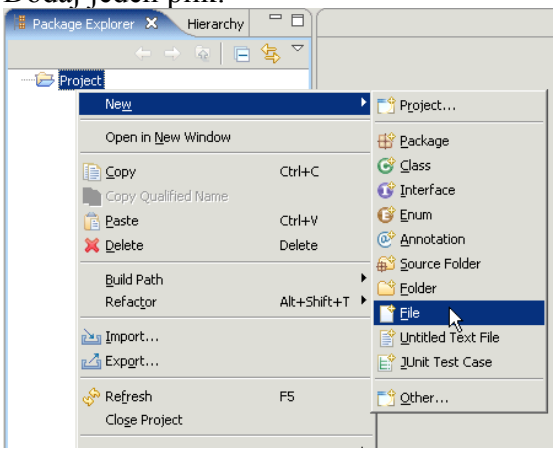
6. Wybierz ogólny projekt



7. Podaj nazwę projektu:

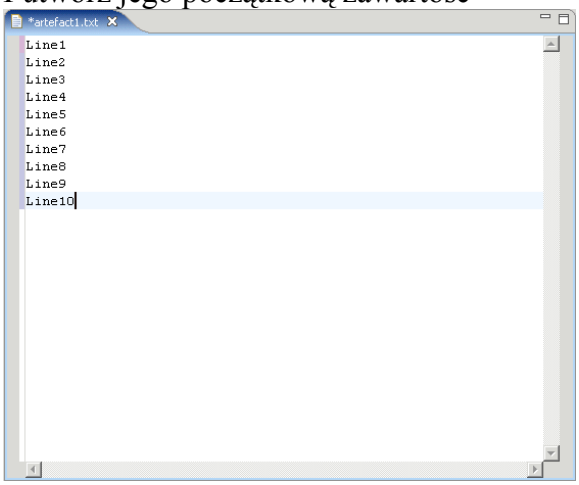


8. Dodaj jeden plik:



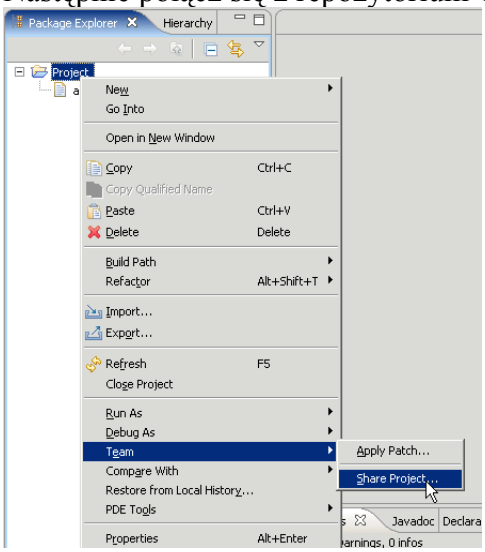
9. Podaj nazwę pliku:

10. I utwórz jego początkową zawartość

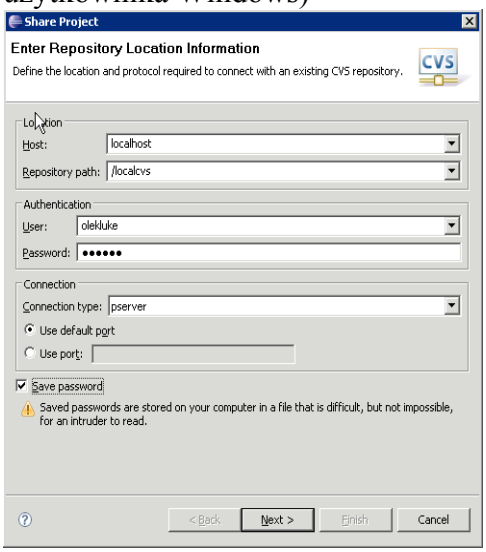


Umieszczenie projektu w repozytorium CVS

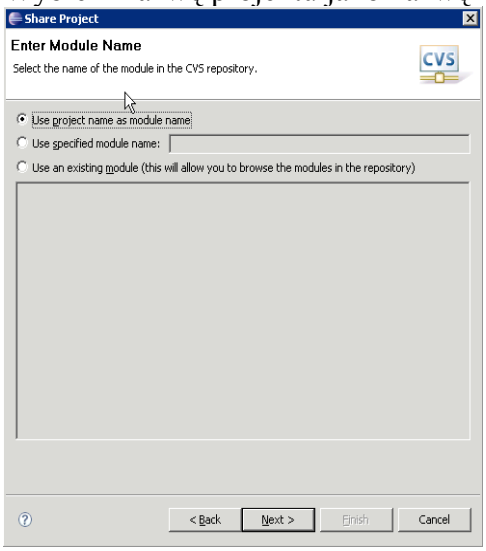
11. Następnie połącz się z repozytorium CVS:



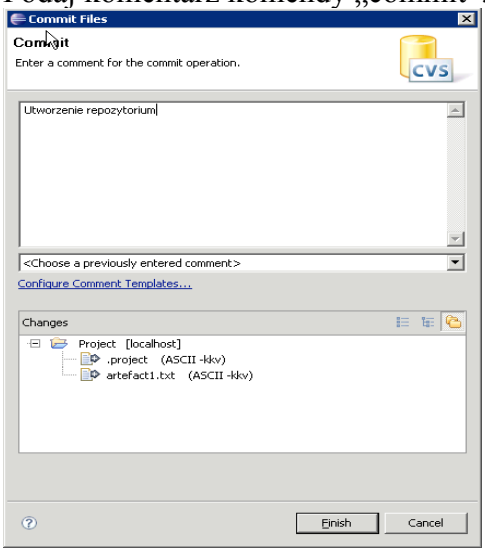
12. Podaj namiary na serwer CVS (jako nazwę użytkownika i hasło – podaj dane użytkownika Windows)



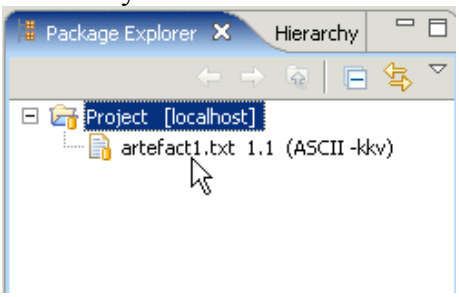
13. Wybierz nazwę projektu jako nazwę modułu z CVS'a



14. Podaj komentarz komendy „commit“:

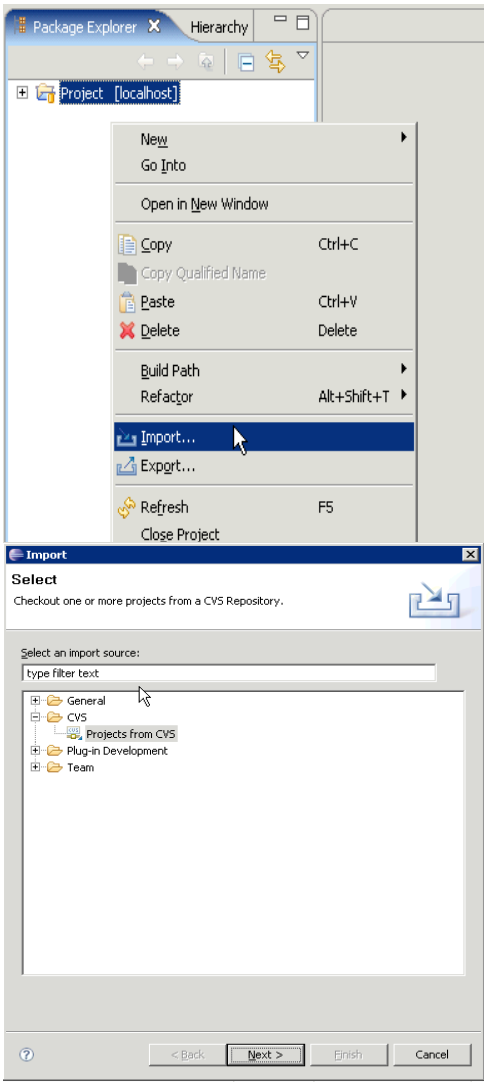


15. Od tego momentu w nawigatorze będą wyświetlane informacje o statusie plików z repozytorium:
    a. W nazwie projektu podana jest lokalizacja repozytorium
    b. Przy nazwie każdego pliku podana jest jego wersja, oraz sposób obliczania zmian (tryb tekstowy – ASCII lub binarny)
    c. Znaczek „>“ na początku nazwy zasobu oznacza, że został on zmieniony lokalnie.

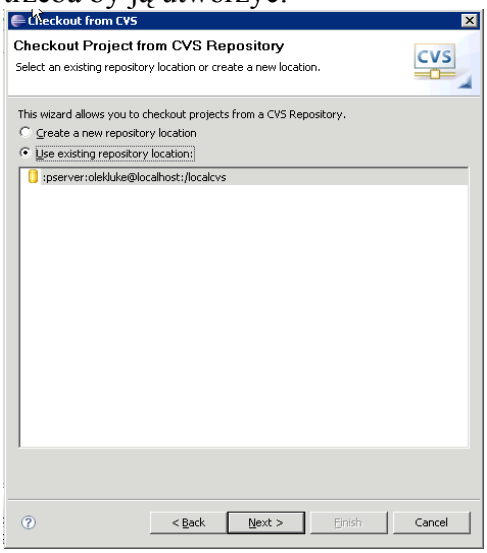


Utworzenie przestrzeni roboczej drugiego programisty

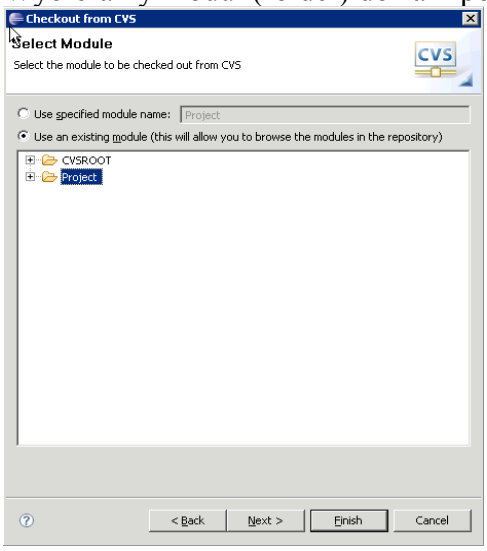
16. Teraz wcielamy się w rolę innej osoby w zespole (my będziemy to robić w ramach jednego środowiska Eclipse – żeby nie opóźniać pracy, lecz normalnie każdy projekt byłby w innym Eclipse na komputerach różnych osób). Tamta osoba po pierwsze pobiera projekt z repozytorium (wykonuje komendę checkout). W Eclipse nazywa się to importem z CVS’a:



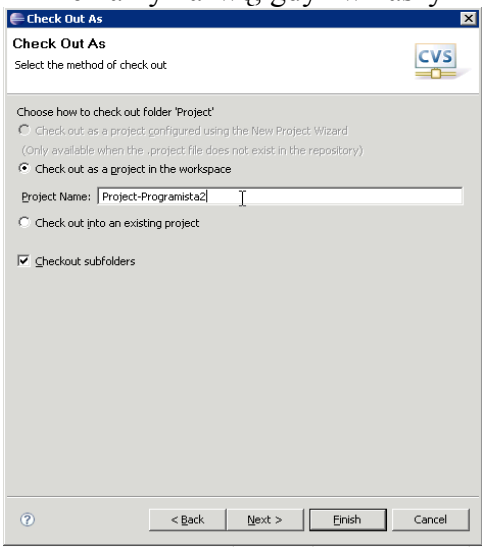
17. W tym przypadku mamy już zdefiniowaną lokalizację CVS'a, lecz normalnie trzeba by ją utworzyć:



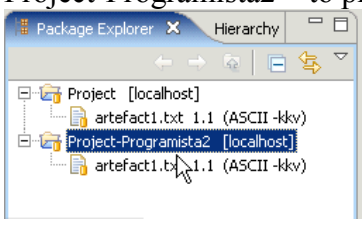
18. Wybieramy moduł (folder) do zaimportowania:



19. I zmieniamy nazwę, gdyż w naszym Eclipse jest już projekt o tej nazwie:

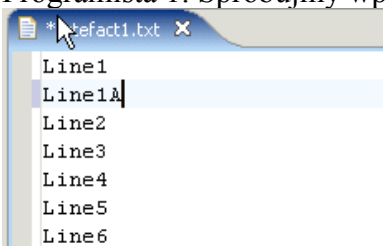


20. W rezultacie otrzymujemy w jednym miejscu projekty 2 programistów:
    a. Project – to przestrzeń robocza pierwszego programisty
    b. Project-Programista2 – to przestrzeń robocza drugiego programisty

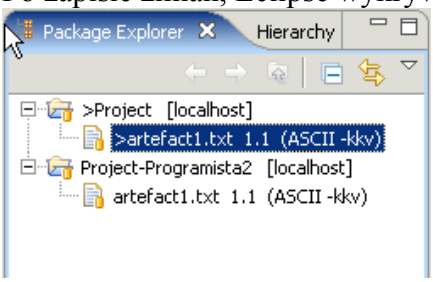


Podstawowa komunikacja poprzez CVS: wysłanie zmian -> odebranie zmian

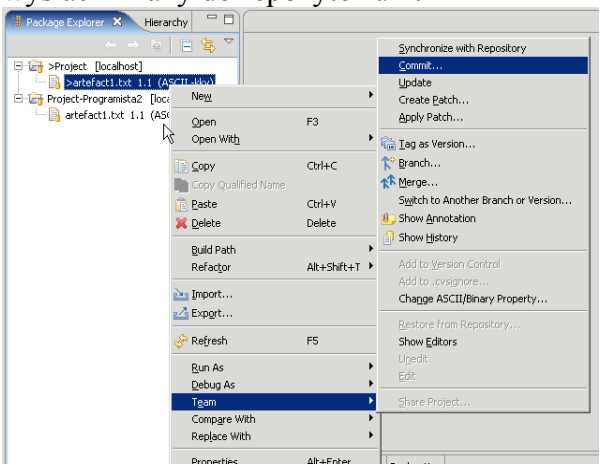
21. Programista 1: Spróbujmy wprowadzić zmiany do pliku. Dodajmy jedną linię:



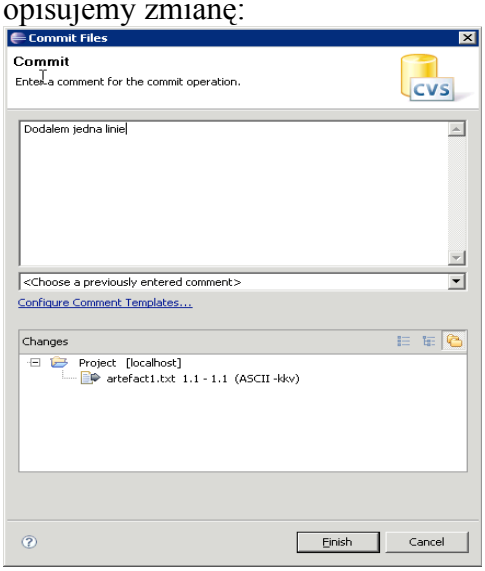
22. Po zapisie zmian, Eclipse wykrywa zmieniony plik i oznacza go „>“.



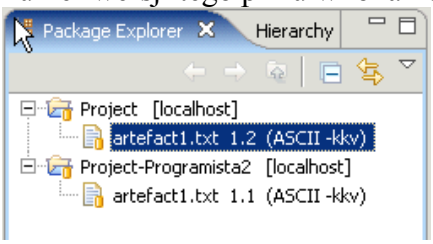
23. Załóżmy, że Programista-1 skończył pracę nad tym artefaktem i pragnie wysłać zmiany do repozytorium:



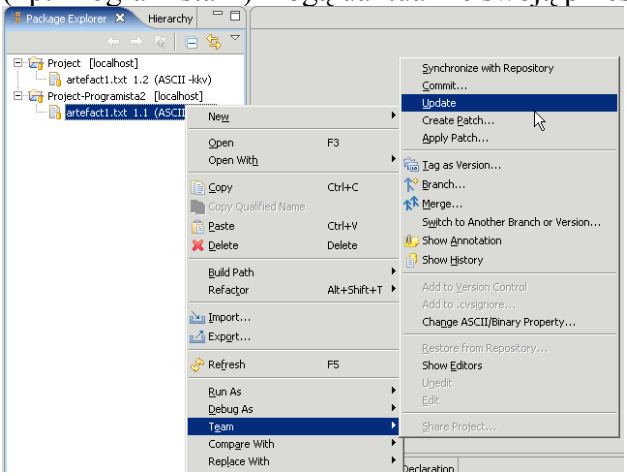
24. Zgodnie z zaleceniami korzystania z CVS, podajemy komentarz, w którym opisujemy zmianę:



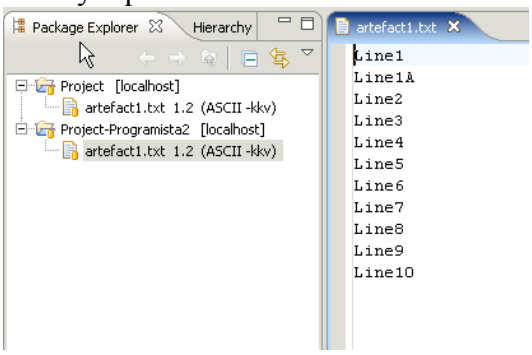
25. Po naciśnięci przycisku Finish, Eclipse wysyła zmiany na serwer i uaktualnia numer wersji tego pliku w lokalnej przestrzeni roboczej Programisty-1:



26. W momencie kiedy mamy zmiany zapisane w repozytorium CVS, inne osoby (np. Programista-2) mogą uaktualnić swoją przestrzeń roboczą:



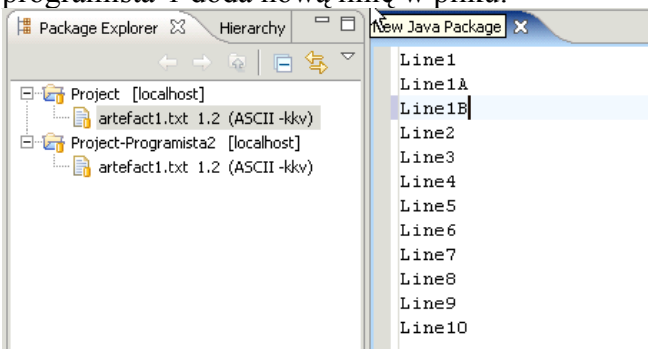
27. Eclipse pobierze zmiany i uaktualni numer wersji tego pliku w przestrzeni roboczej Programisty-2. Możemy się upewnić, czy zmiany rzeczywiście zostały wprowadzone:



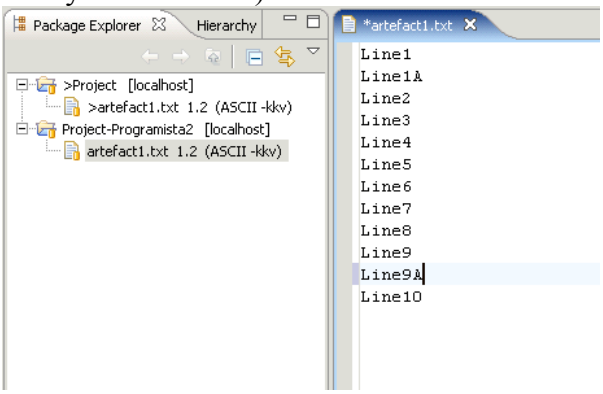
Równoległa praca nad zasobem – brak konfliktu

28. Zasymulujmy teraz równoległą pracę nad zasobem. Ponieważ pracujemy tylko na jednym komputerze, więc będziemy musieli zrobić to sekwencyjnie, lecz z punktu widzenia serwera CVS będzie to praca równoległa. Najpierw niech

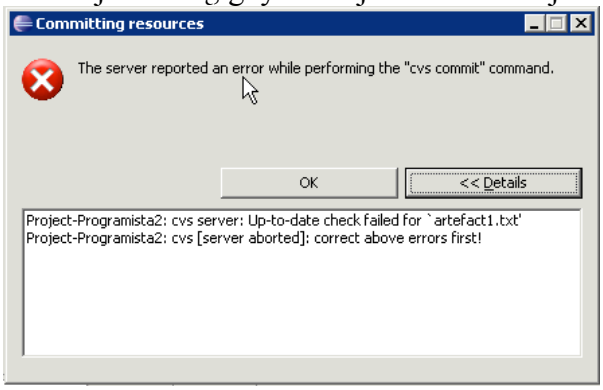
programista-1 doda nową linię w pliku:



29. Następnie programista-2 niech doda swoją linię w innym miejscu pliku (aby nie było konfliktów):

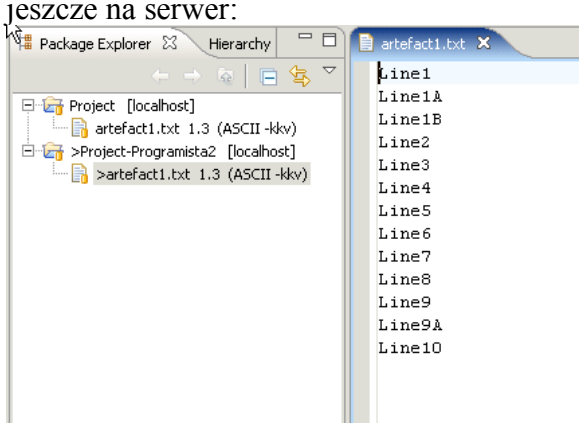


30. Programista-1 wysyła swoje zmiany (polecenie Commit, kroki 23-24)
31. Programista-2 próbuje wysłać swoje zmiany (również polecenie Commit, lecz tym razem na pliku z projektu Project-Programista2, kroki 23-24). Niestety nie udaje mu się, gdyż wersja na serwerze jest nowsza niż wersja lokalna:

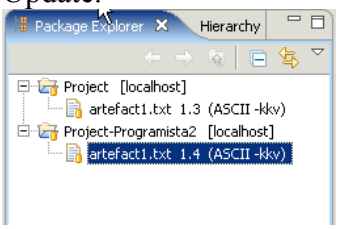


32. Należy więc najpierw wykonać komendę „Update“ (krok 26) – Eclipse ściągnie zmiany i uaktualni wersję do 1.3. Proszę zauważyć, że teraz plik ten będzie zawierał zmiany wprowadzone wcześniej lokalnie (Line9A, jak również wprowadzone wcześniej przez Programistę-1: Line1B). Plik nadal jest oznaczony symbolem „>“, gdyż posiada lokalne zmiany, nie wysłane

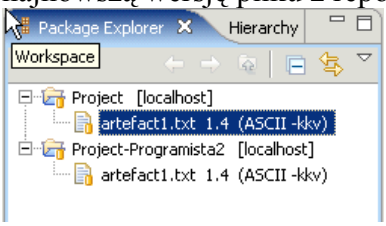
jeszcze na serwer:



33. Teraz Programista-2 może wreszcie wykonać komendę Commit, która tym razem wykona się bez błędów. CVS nada nowy numer wersji (1.4). Programista-1 nadal ma wersję 1.3, gdyż nie wykonał jeszcze komendy Update.

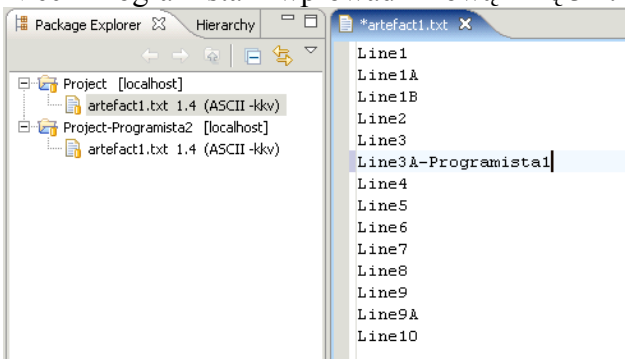


34. Po wykonaniu komendy Update przez Programistę-1, oboje będą mieli najnowszą wersję pliku z repozytorium:

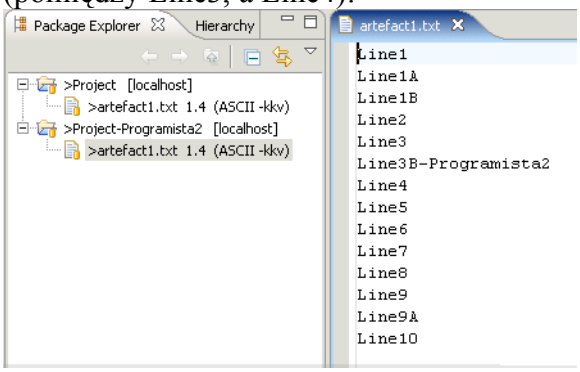


Równoległa praca przez 2 programistów – konflikt w pliku

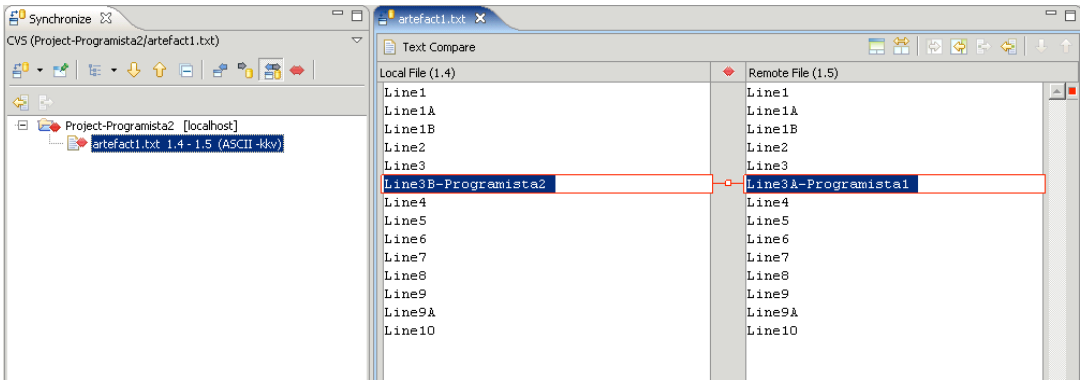
35. Niech Programista-1 wprowadzi nową linię 3A:



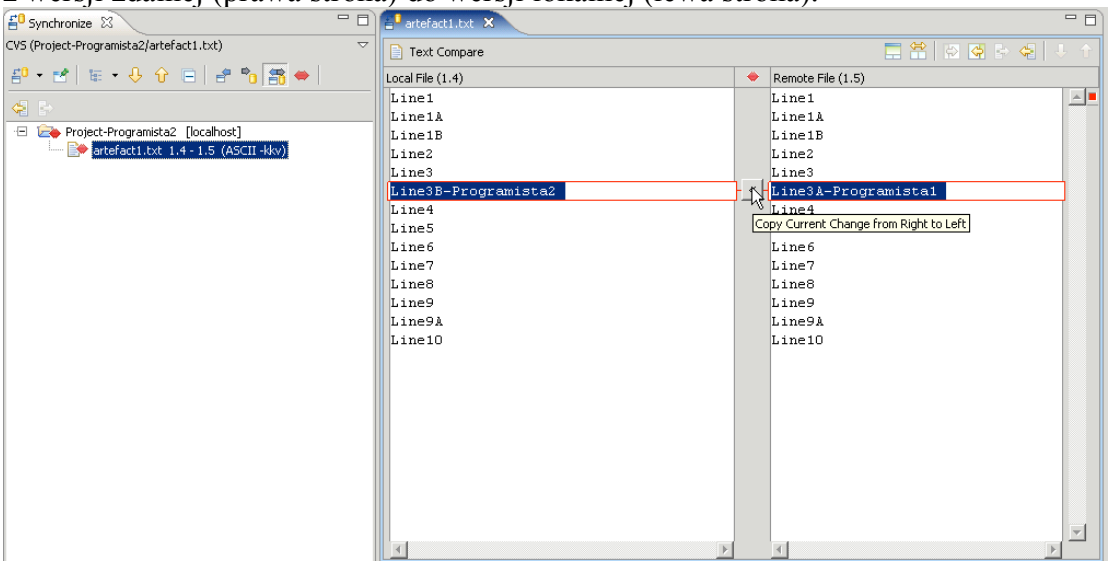
36. Również Programista-2 niech doda nową linię w tym samym miejscu (pomiędzy Line3, a Line4):



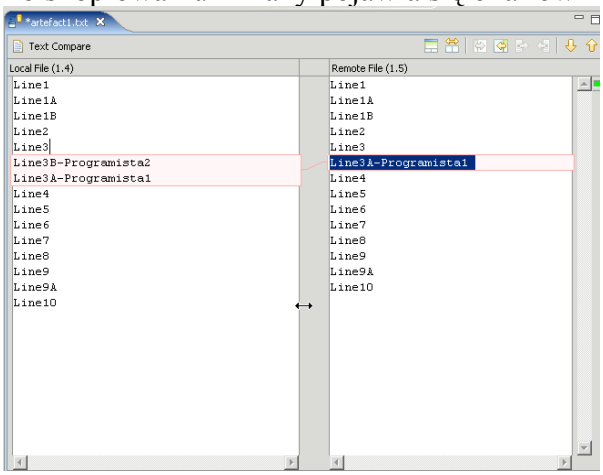
37. Po wysłaniu zmian przez Programistę-1 (polecenie Commit) są one już widoczne w repozytorim (na razie CVS nie wie, że Programista-2 również uaktualnił to samo miejsce – czyli że będzie konflikt). W momencie kiedy Programista-2 chce wykonać komendę Update okazuje się, że jest konflikt. Eclipse przełącza się do specjalnego widoku rozwiązywania problemów z CVS i umożliwia otwarcie edyta konfliktów:



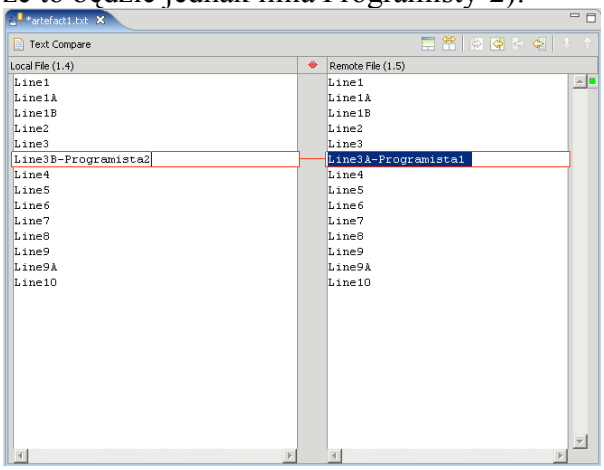
38. Miejsca konfliktowe są zaznaczone kolorem czerwonym. Zadaniem użytkownika jest teraz utworzenie wersji spójnej, czyli przejrzenie wszystkich konfliktowych zmian i wybranie poprawnej wersji. Przy każdym konflikcie znajduje się przycisk „<“, za pomocą którego można daną zmianę skopiować z wersji zdalnej (prawa strona) do wersji lokalnej (lewa strona).



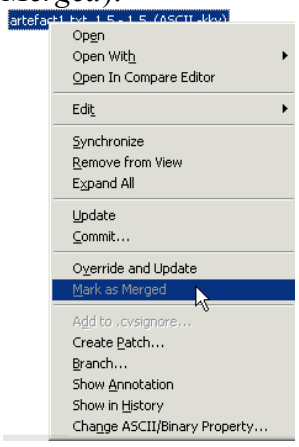
39. Po skopiowaniu zmiany pojawia się ona również w wersji lokalnej:



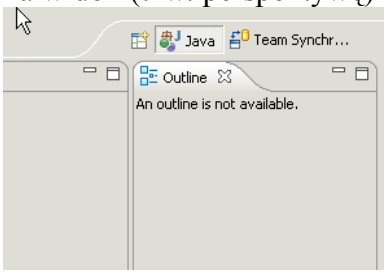
40. Programista może teraz dokonać edycji i zostawić poprawną wersję (załóżmy, że to będzie jednak linia Programisty-2):



41. Tak przygotowany plik należy następnie oznaczyć jako „połączony“ (ang. *Merged*):



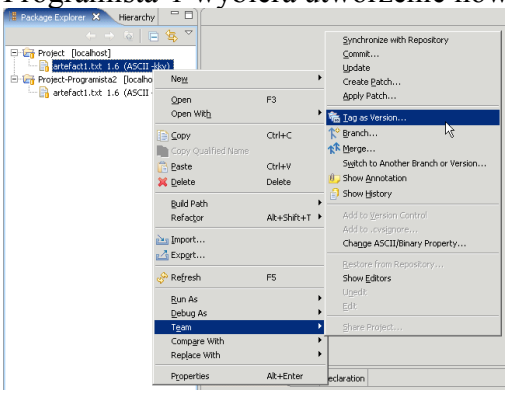
42. Po zakończeniu rozwiązywania konfliktów można przełączyć się spowrotem na widok (tzw. perspektywę) edycji kodu Javy:



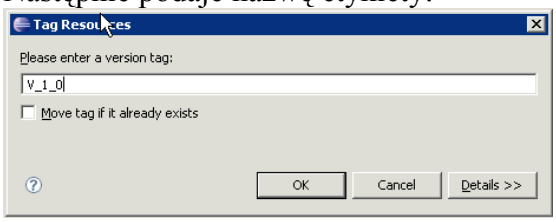
43. Teraz Programista-2 może już spokojnie wysłać swoją wersję do repozytorium (komenda Commit)

Praca z etykietami (tags)

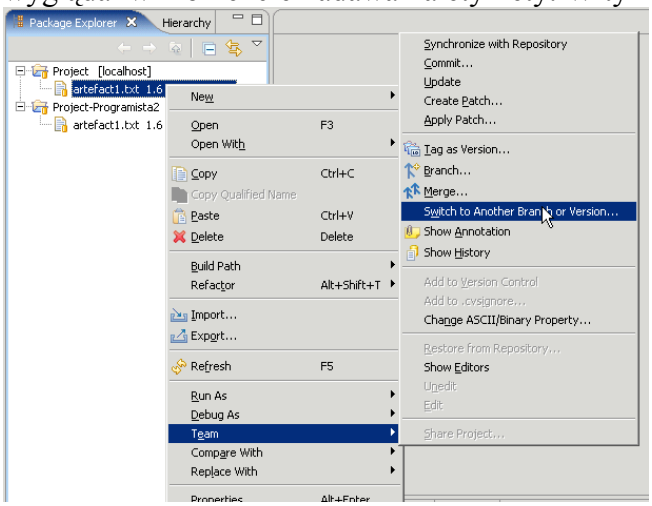
44. Spróbujmy nadać fragmentowi projektu (jednemu plikowi) etykietę. Programista-1 wybiera utworzenie nowej etykiety:



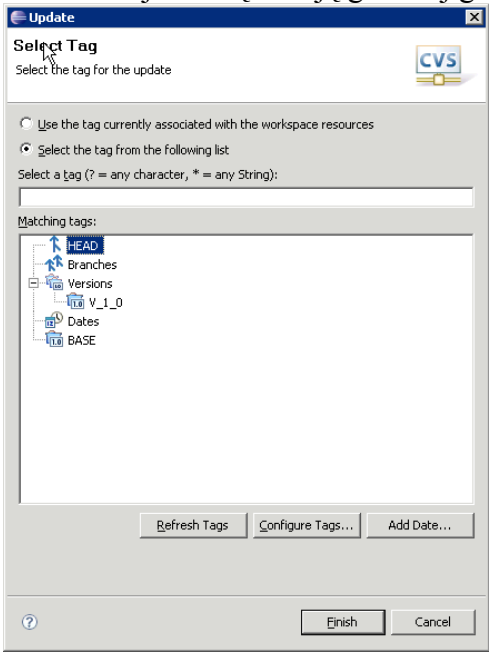
45. Następnie podaje nazwę etykiety:



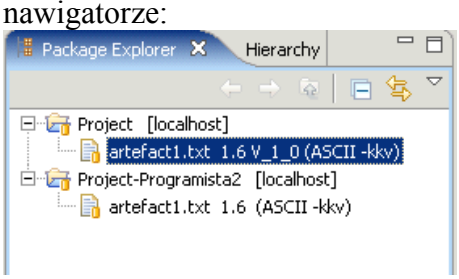
46. Etykieta została nadana. Załóżmy również, że w pewnym momencie, po wprowadzeni wielu zmian na tym zasobie chcemy wrócić i zobaczyć, jak plik wyglądał w momencie nadawania etykiety. W tym celu należy:



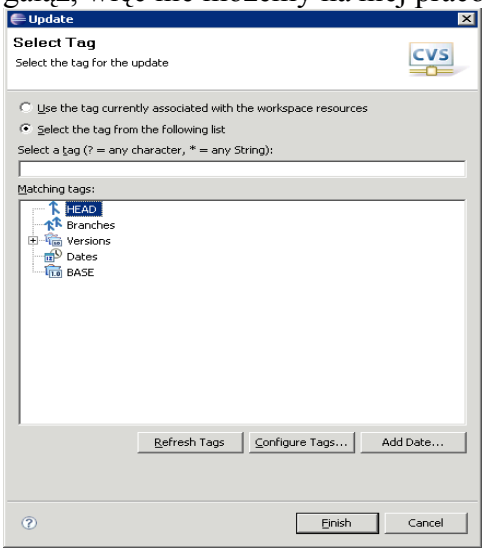
47. Pojawi się lista wszystkich możliwych etykiet do wyboru. Etykieta HEAD oznacza najnowszą wersję głównej gałęzi projektu:



48. Po pobraniu wersji pliku z podanej etykiety Eclipse oznacza to odpowiednio w nawigatorze:



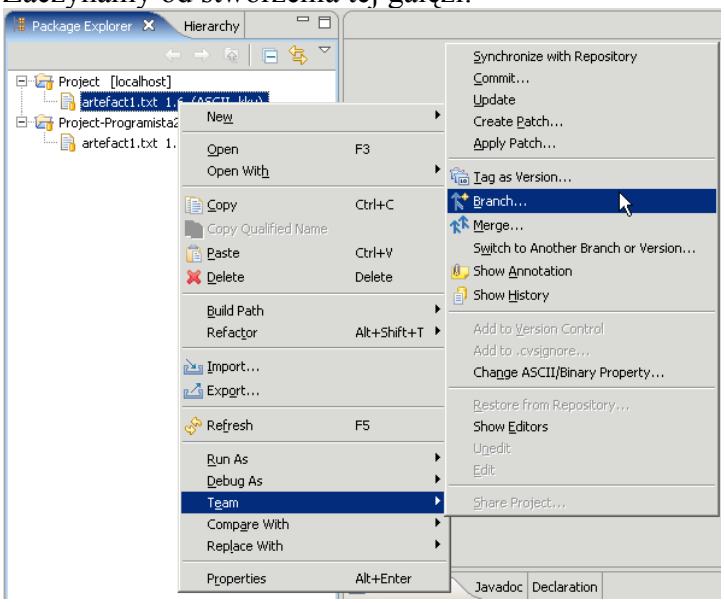
49. Jeżeli chcemy kontynuować pracę nad tym artefaktem, musimy przełączyć się z powrotem do najnowszej wersji danego pliku (HEAD) (to jest etykieta, nie gałąź, więc nie możemy na niej pracować równolegle do głównej linii):



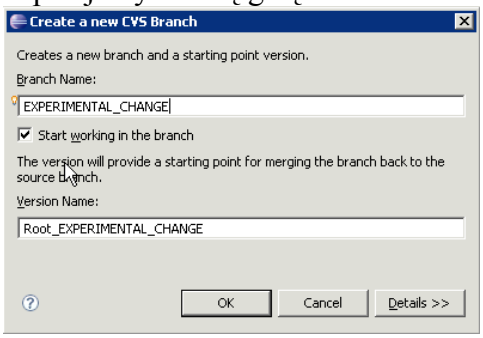
Rozgałęzianie prac

50. Programista jeden ma do wykonania duże zadanie, które w mocny sposób ingeruje w strukturę kodu. Najlepiej jest to zrobić w osobnej gałęzi.

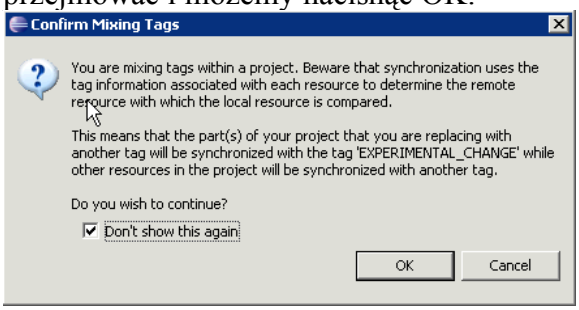
Zaczynamy od stworzenia tej gałęzi:



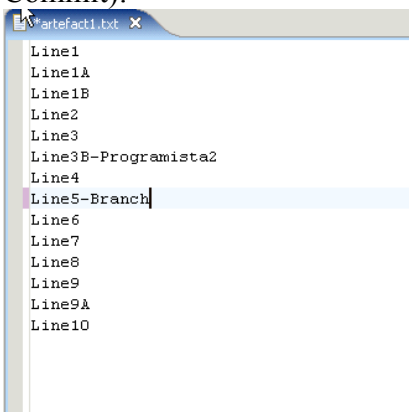
51. Wpisujemy nazwę gałęzi:



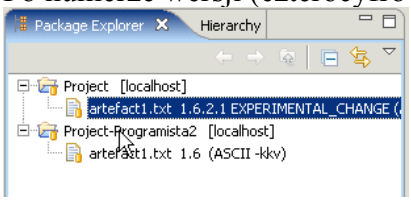
52. Eclipse ostrzega nas, że w ramach jednego projektu mieszamy gałęzie (tworzymy gałąź tylko dla jednego pliku z projektu). Nie musimy się tym przejmować i możemy nacisnąć OK:



53. Następnie wprowadzamy zmiany do pliku i wysyłamy je na serwer (polecenie Commit):

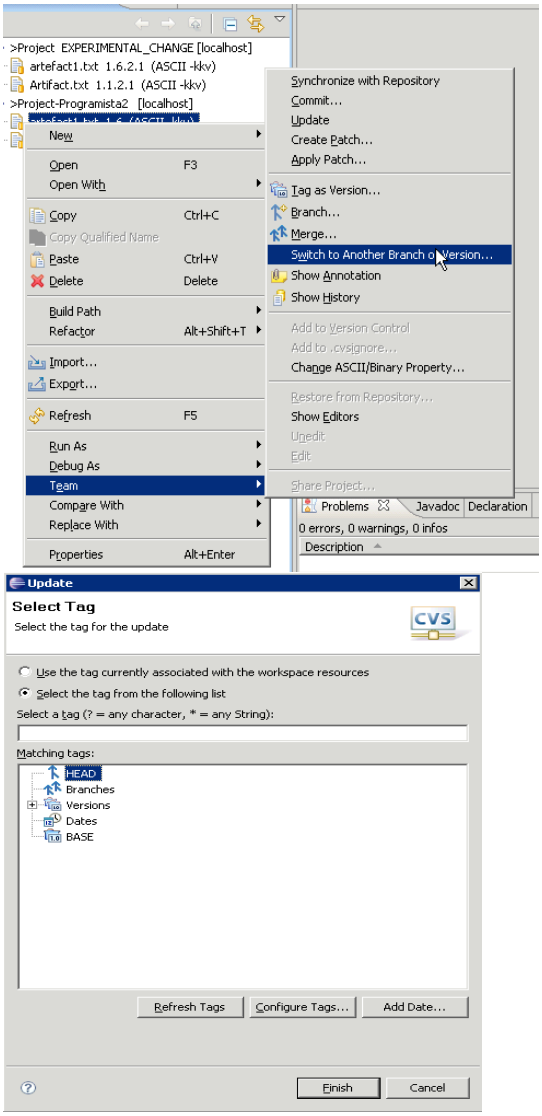


54. Po numerze wersji (czterocyfrowy) możemy poznać, że pracujemy w gałęzi.

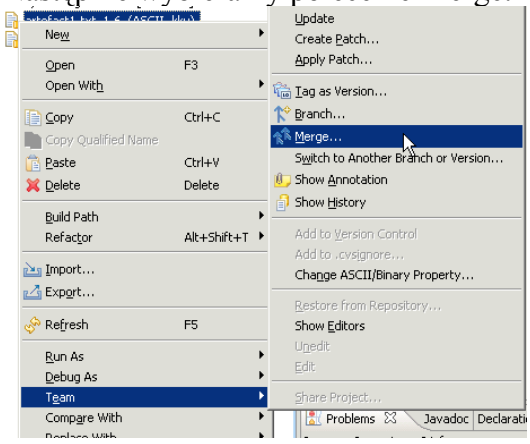


Scalanie zmian do gałęzi głównej

55. W pierwszej kolejności należy się przełączyć do gałęzi, do której chcemy scalić zmiany (w naszym przypadku będzie to HEAD):

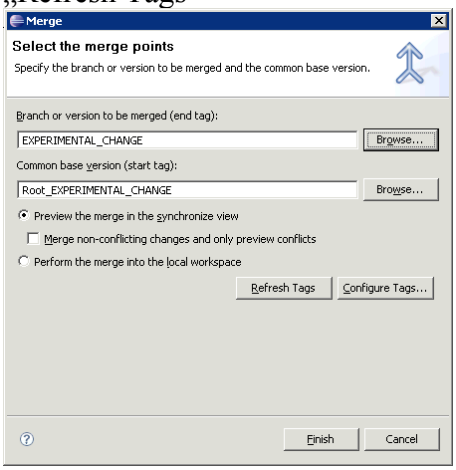


56. Następnie wybieramy polecenie Merge:

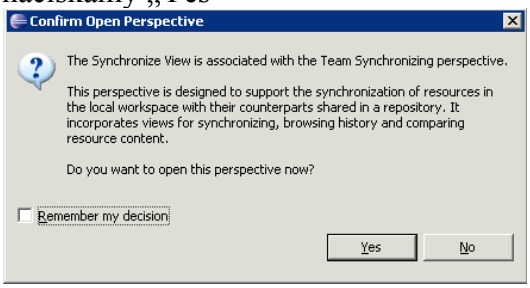


57. A następnie gałąź, z której zmiany chcemy scalić z główną gałęzią. Jeżeli by się zdażyło, że naszej gałęzi nie można wybrać, to należy wykonać polecenie

„Refresh Tags“



58. Eclipse zapyta, czy może się przełączyć do widoku synchronizacji z CVS, naciskamy „Yes“



59. Po czym, w widoku podobnym do rozwiązywania konfliktów pokazuje zmiany pomiędzy gałęzią główną, a gałęzią EXPERIMENTAL_CHANGE. Można wybrać, które zmiany chcemy przekopiować do gałęzi głównej. Dalej postępujemy dokładnie tak, jak w przypadku rozwiązywania konfliktów (oznaczamy plik jako „Merged“ i robimy „Commit“ zmian).