Projet de développement

Introduction à Eclipse

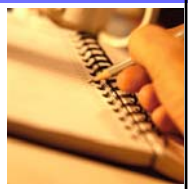
Philippe Collet

Licence 3 Informatique

2011-2012

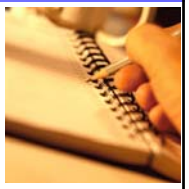
Université Nice SOPHIA ANTIPOLIS

Organisation



- **Cours 1 : principes généraux - svn**
- **Cours 2 : Redmine et gestion de projet**
- **Cours 3 : Introduction à Eclipse**
- **Cours 4 : Eclipse C / PHP, V&V et tests unitaires en Java**
- **Cours 5 : Tests C / PHP**
- **Cours 6 : conclusion , questions…**

Plan



- **Application de redmine à votre projet**
- **Eclipse : Introduction et historique**
- **Architecture d'Eclipse**
- **Environnement de développement**
- **JDT**
- **Support SVN**
- **Application à votre projet**

Application à votre projet

Université Nice SOPHIA ANTIPOLIS

Redmine : application à votre projet

- **Pour votre projet :**
  - Démo en ligne modifiable :
    - http://demo.redmine.org
    - Essai du wiki, modification de page, différentes fonctionnalités…
- **Dès que votre redmine de projet est disponible**
  - Inscription (ou vérification d'inscription) de chaque membre
  - Création d'un milestone « prise en main »
    - Création d'un ticket « prise en main » pour chaque membre
    - Ecriture d'un commentaire relatif au ticket, et au milestone
    - Fermeture et validation de son ticket par chaque membre
  - Ajout de fichiers (du projet ou d'essai) dans le subversion
    - Modification des fichiers / nouveau commit (ligne de commande ou eclipse/subclipse)
    - Navigation dans le subversion par redmine
    - Création de ticket « bidon » et association à un commit fait sur le subversion !

Redmine : application à votre projet

- **Par la suite**
  - Première séance
    - Découpage du travail en components et milestones
    - Création de tickets dans les milestones pour les tâches
    - Attribution des tâches et validation par le tuteur
  - Autres séances
    - Fermeture/Ouverture de tickets en fonction des commits, des tests
- **La vision « redmine » résultante du projet fait très largement partie de votre évaluation**
  - Qui fait quoi
  - Traçabilité tout au long du projet

La roadmap peut être utilisée pour la soutenance…

Eclipse

Université Nice SOPHIA ANTIPOLIS

Introduction : objectif

- **Un Environnement de Développement Intégré (EDI)**
  - Un logiciel regroupant un ensemble d'outils nécessaires au développement des applications dans un langage de programmation
- **Objectifs généraux : fournir des fonctionnalités**
  - un éditeur de texte spécialisé
  - un compilateur
  - un débogueur
  - des outils automatiques de gestion d'applications ayant plusieurs fichiers source (projets)
  - un gestionnaire de versions
  - un générateur de documentation

Historique

- **Préhistoire :**
  - 1950 : cartes perforées
  - 1960 : terminaux, éditeurs de texte basique, compilateur et débogueur en ligne de commande
  - 1970 : introduction des makefiles et des fichiers de configurations
- **1980 : environnement graphique et premiers EDI (1981 Turbo Pascal)**
  - 1983 : Borland Turbo Pascal (DOS) à 50 $
  - 1987 : Borland Turbo C
  - 1991 : Microsoft Visual Basic 1
  - 1997 : Microsoft Visual Studio (C++)

Eclipse et ses concurrents

- **Logiciels libres :**
  - Emacs, XEmacs : basique, mais adaptables à tout langage
  - OpenOffice.org : langages de script
  - Kdevelop (KDE) : C, C++, basé sur les outils GNU
  - Netbeans (Sun/Oracle) : initialement conçu pour Java, maintenant C, C++, XML et HTML
    - Au départ plus lourd et lent, maintenant plus léger (et vraisemblablement plus rapide)
  - Eclipse (OTI-IBM) : Java, C/C++, PHP, HTML, etc.
- **Logiciels propriétaires :**
  - Visual Studio (Microsoft) : C/C++, .NET, C#, etc.
  - JBuilder (Borland) : Java
    - Abandonné au profit de… plugins Eclipse !
  - JCreator : Java
  - WinDev (PC Soft) : application PC Pocket et Mobile

Eclipse

- **Logiciel libre**
  - Destiné à l'origine pour le développement en Java
  - Conçu sur la base d'un EDI Java (VA4J), Eclipse devient un EDI pour développer des EDIs et d'autres outils
- **Objectif :**
  - Offrir une plateforme ouverte pour le développement d'applications
  - Non dédiée à un langage ou système d'exploitation ou interface graphique
  - Facile à comprendre mais aussi facile à étendre
  - Paramétrable selon les besoins/goûts du programmeu
  - Capable d'automatiser les taches lourdes du développement
  - Ayant une base stable
  - Utilisable pour son propre développement (bootstrap-able)
  - Promouvant l'utilisation de Java

Historique d'Eclipse

- 1996 : IBM rachète OTI, qui développe la suite d'EDI Visual Age (en SmallTalk), et en particulier VA4J
- 2001 : après un investissement de 40 M$, IBM lance Eclipse 1
  - Grand succès populaire car suite ouverte et gratuite (licence CPL).
  - Création du consortium Eclipse (IBM, Borland, RedHat, SuSE, Intel,...)
- 2002 : Eclipse 2.0
- 2004 : Eclipse 3.0
- 2006 : Eclipse 3.2 Europa
  - Première release nommée pour stabiliser les références aux sous-projets
- 2009 : Eclipse 3.5 Galileo
- 2011 : Eclipse 3.7 Indigo

Sous-projets Eclipse

- **Eclipse : architecture et structure de la plateforme**
- **Eclipse Tools : outils pour permettre l'enrichessement de la plateforme**
  - PDT, CDT sont basés sur ce sous-projet
- **Eclipse Technology : recherche sur l'évolution de la plateforme**
  - Très actif pour le passage de Eclipse 2.x à Eclipse 3.x
- **Test and Performance Tools Platform (TPTP) : outils de test et d'analyse**
- **Business Intelligence and Reporting Tools (BIRT) : outils de génération d'états**
  - Composé de 4 autres sous-projets

Sous-projets Eclipse

- **Eclipse Modeling : Plusieurs sous-projets dont**
  - EMF (Eclipse Modeling Framework) : pour la manipulation de modèles et projection vers du code
  - UML2 : métamodèle complet d'UML2 pour création d'outils conforme
- **Data Tools Platform (DTP) : Manipulation de source de données (BD relationnelles essentiellement)**
- **Device Software Development Platform : Outils pour plugins de développement dédié aux applications mobiles**
- **Eclipse SOA Tools Platform (STP) : Outil pour le développement d'applications selon des architectures orientées services (web services, standard SCA…)**

Installation

- **Simplissime :**
  - Téléchargez l'archive (dédiée au système ou générique)
  - Décompactez la dans un répertoire système
  - Créez un lien/raccourci vers l'exécutable eclipse…
  - Ca roule
- **Un peu moins simple : quels plugins et comment les installer ?**
  - Décompactez le plugin dans le répertoire dédié
  - Utilisez la fonction d'update (tutoriaux en ligne)
- **Moins simple : quelle version prendre ?**
  - Classic
  - J2EE (Java Entreprise…)
  - Etc.

Votre installation

- **Page web : http://deptinfo.unice.fr/twiki/bin/view/Linfo/ProjetDev2012Outils**
- **Java**
  - Eclipse 3.7 JEE edition : http://www.eclipse.org/downloads/
  - plugin subclipse 1.8 (accès au référentiel svn) : http://subclipse.tigris.org/update_1.8.x (par update)
- **PHP/MySQL**
  - PDT 3.0 http://www.eclipse.org/pdt/downloads/ : envt standard PHP dans Eclipse
  - plugin subclipse 1.8 (accès au référentiel svn) : http://subclipse.tigris.org/update_1.8.x (par update)
  - SimpleTest (ne pas utiliser le plugin) : http://www.simpletest.org/fr/start-testing.html
- **En cours d'installation…**

Constituants

- **Eclipse = plateforme + plug-ins**
- **Plateforme**
  - Un support d'exécution (runtime) indépendant du système d'exploitation (JVM)
  - Un ensemble basique de plug-ins extensibles
  - De mécanismes (API), règles et outils pour construire de plug-in
  - Un moteur pour découvrir, charger et exécuter des plug-ins
- **Plug-in = la plus petite unité qui peut être développée et utilisée séparément**
  - se connecte à un point précis de la plateforme
  - remplit une tache (pas forcement exécutable)
  - offre des points d'extension
  - coexiste avec d'autres plug-ins
- **instance (feature) = ensemble de plug-ins qui coopèrent pour offrir un EDI**

Eclipse : support d'exécution

- **Support d'exécution = Platform Runtime**
  - Exécute la JVM (Java Virtual Machine)
  - Définit les points d'extension et le modèle plug-in
- **Modèle de plugin**
  - point d'extension = interface
  - plug-in = interfaces implémentées + archive Jar + interfaces utilisées
  - déclaration de plug-in = manifeste (dépendences à l'exécution) + interface (type)
- **Le support d'exécution**
  - Découvre dynamiquement les plug-ins et maintient une base relative à leur déclaration
  - Charge les plug-ins à la demande.
  - Met à jour automatiquement des instances (features)

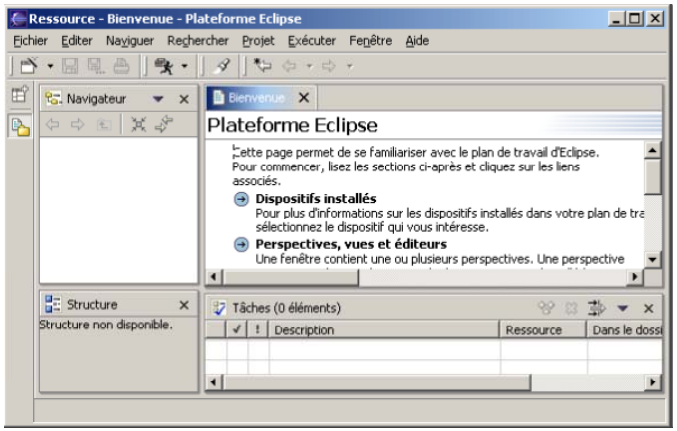
Environnement de développement

Université Nice SOPHIA ANTIPOLIS

Plan de travail

- **Plan de travail = Workbench**
  - Fournit l'interface visuelle pour l'utilisateur de la plateforme
  - (Spécificité Eclipse) : l'interface graphique (UI) a l'apparence d'une application native du système d'exploitation
  - est basé sur deux outils (SWT – Standard Widget Tool, JFace) qui peuvent être utilisés directement pour développer des applications
- **Composantes physiques de l'UI : menus, barre d'actions, boutons, onglets, fenêtres**
- **Composantes logiques de l'UI (paramétrable par des plug-ins)**

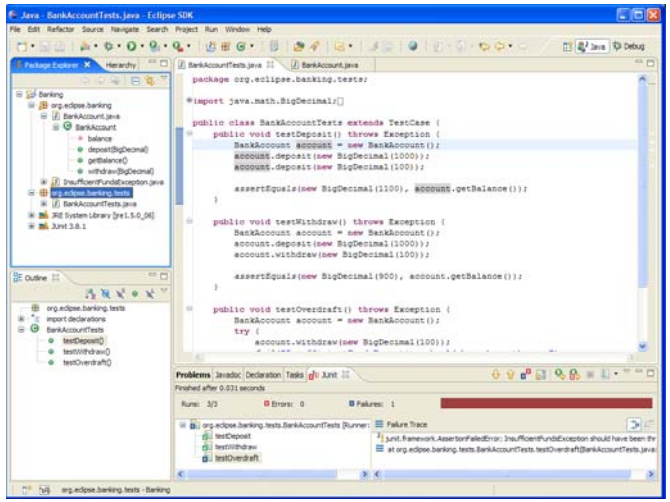
Workbench



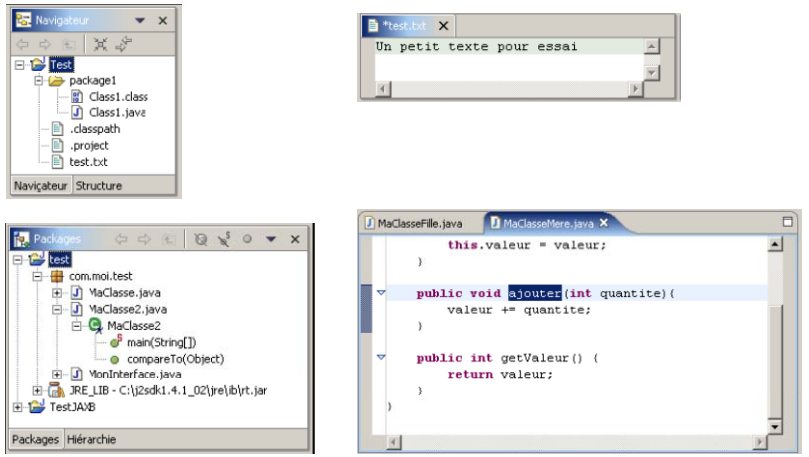
Vues et perspectives

- **Vue**
  - fournit des informations sur les objets (structure, composantes, etc.) en communiquant avec d'autres vues ou éditeurs
  - Exemple: Navigateur, packages
- **Editeur**
  - Edition plus ou moins dédié (langage avec coloration syntaxique, complétion)
  - Exemple : éditeur de texte, éditeur Java
- **Perspective**
  - ensemble d'éditeurs et vues ayant une disposition précise dans le plan de travail
  - Afin de faciliter la réalisation de certaines taches
  - Exemple : navigation, édition Java, exploration de référentiel cvs/svn, synchronisation avec un référentiel
  - Le plus étendu des points d'extension

Perspective

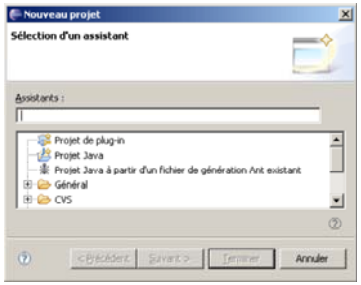


Vues et éditeurs



Assistants

- **Assistant**
  - Facilite la saisie et/ou sélection d'information par des panneaux successifs
  - Très nombreux et très utilisés

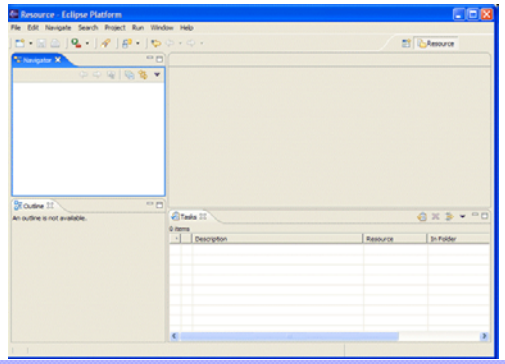


L'espace de travail : Workspace

- **Workspace**
  - Ressources : fichiers, répertoires, projets, etc.
  - Espace de travail = un ou plusieurs projets
- **Projet = partie du système de fichiers qui a une personnalité (définie par les plug-ins)**
  - Exemples : projet Java, site Web.
- **Le workspace implémente un mécanisme d'historique locale (backup) pour tracer les changements des ressources**

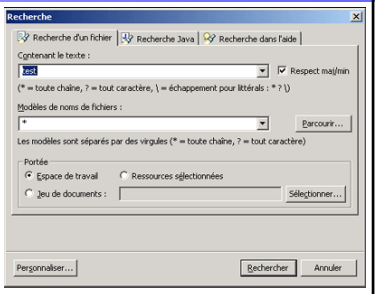
Perspective Ressource

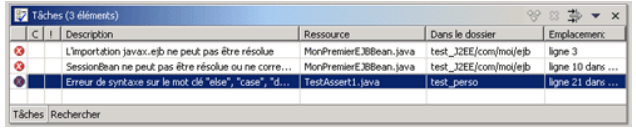
- **Par défaut, cette perspective contient les fenêtres suivantes :**
  - la vue "Navigateur" qui affiche les ressources (arborescence des fichiers) de l'espace de travail
  - un éditeur qui permet d'éditer une ressource sélectionnée dans la vue "Navigateur"
  - la vue "Structure" qui permet d'obtenir une arborescence présentant les grandes lignes de certaines ressources en cours de traitement
  - la vue "Tâches" qui affiche une liste de tâche à effectuer



Fonctions pratiques

- **Recherche**
  - Dans tout l'espace
  - Dans des fichiers
  - Une recherche dédiée à Java
- **Taches**
  - actions à réaliser
  - erreurs de compilation à corriger
  - points d'arrêt pour le débogage





Aide en ligne

- **F1 : aide contextuelle**
  - Dépend de la vue, l'éditeur, etc.
- **Dans un éditeur :**
  - CTRL + ESPACE => complétion

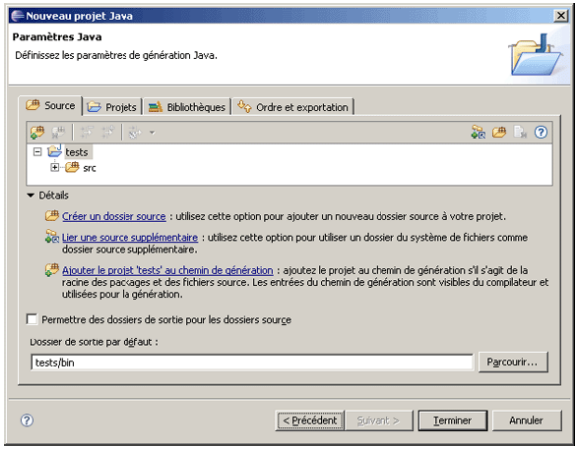
JDT

Université Nice SOPHIA ANTIPOLIS

JDT : Java Develoment Tooling

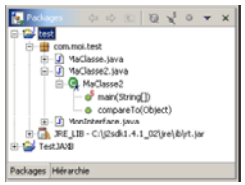
- **Composants**
  - les perspectives "Java" et "Navigation Java"
  - les vues "Packages" et "Hierarchie"
  - les éditeurs "Java" et "Scrapbook"
  - les assistants : pour créer de nouveaux projets, packages, classes, interfaces, ...
- **Outils de création**
  - Projet
  - Packages
  - Classes
  - Interfaces

Projet Java

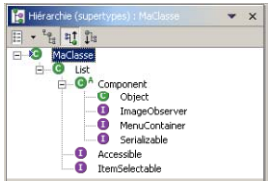


Vues du JDT

- **Package**
  - Arborescence du code source



- **Hiérarchie**



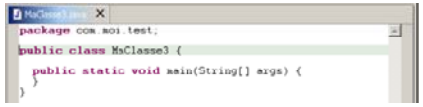
- **Javadoc, déclarations, erreurs, historique**

Editeur de code

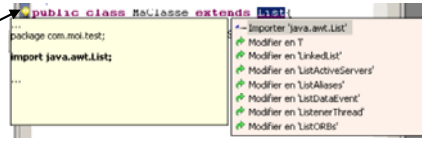
- **Spécifique Java avec les fonctionnalités suivantes**
  - coloration syntaxique
  - complétion de code (CTRL+ESPACE)
  - formatage du code source (CTRL+SHIFT+F)
  - l'importation et l'exportation de code via un assistant
  - forte synergie avec le débogueur

Editeur de code

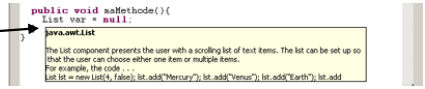
- **Coloration syntaxique**



- **Proposition de correction**

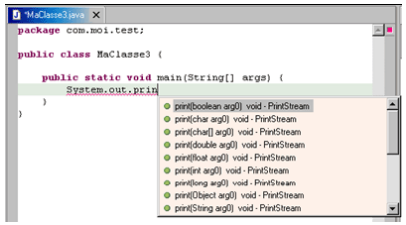


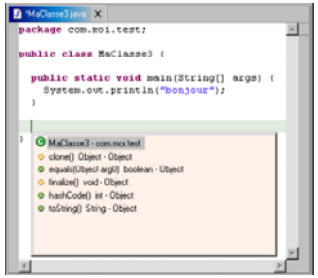
- **Bulle d'aide**



- **Formatage, masquage de portion de code, etc.**

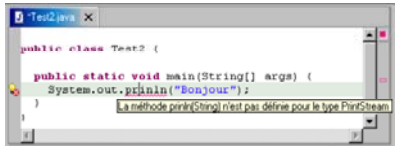
Complétion

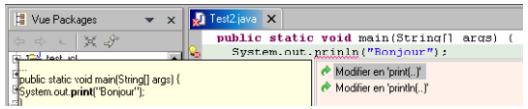




Compilation et erreur

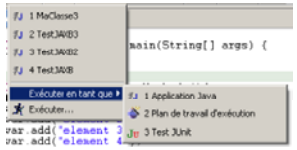
- **Par défaut**
  - La compilation se fait « tout le temps », en arrière plan
- **Erreurs**



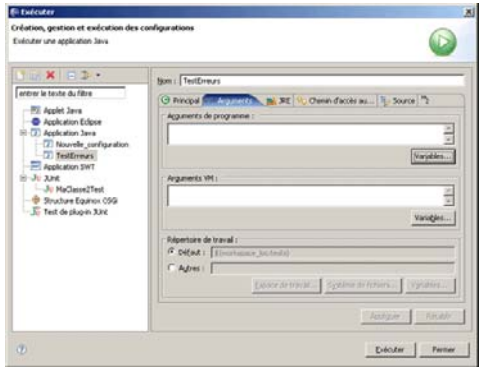


Exécution

- **Sur une classe :**



- **Par des configurations**
  - Classe
  - Paramètre
  - JRE
  - Bibliothèques
  - CLASSPATH
  - …



Refactoring

- **Structure du code**
  - Renommer, déplacer
  - Changer la signature d'une méthode
  - Encadrer (try/catch)
- **Structure au niveau de la classe**
  - Transférer/Extraire méthode ou attribut
  - Extraire une interface
- **Structure à l'intérieur d'une classe**
  - Intégrer méthode/attribut
  - Extraire une méthode, variable locale, constante

CVS/SVN dans Eclipse

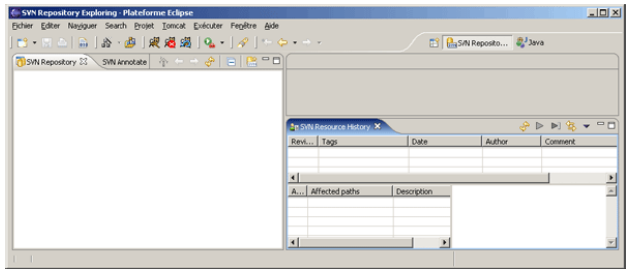
Université Nice SOPHIA ANTIPOLIS

Gestion de versions

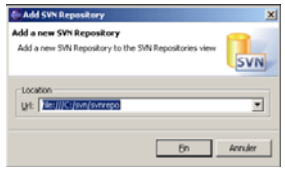
- **Team support = gestion de versions**
  - Contrôle les versions et le partage d'un projet entre différents développeurs
  - enregistre dans une archive
  - gère des modifications de fichiers
  - récupère toute modification enregistrée
  - visualise les différences entre les versions
- **CVS (Concurrent Version System) est utilisé par défaut**
- **Eclipse fournit une API pour l'interface avec d'autres systèmes**
- **Subclipse : Support SVN dans Eclipse**
  - *subclipse.tigris.org*
- **Subversive : support en standard dans Eclipse**
  - Finalement moins stable que Subclipse, à éviter pour l'instant…

Perspective « SVN Repository Exploring »

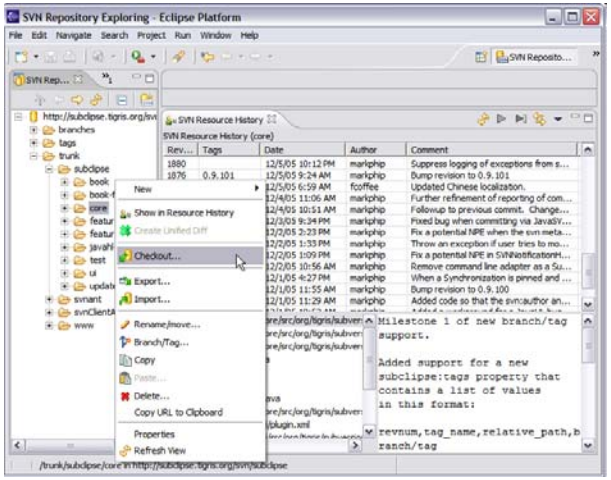
- **Perspective pour administrer les différents référentiels svn utilisés**



- **Ajout d'un référentiel**

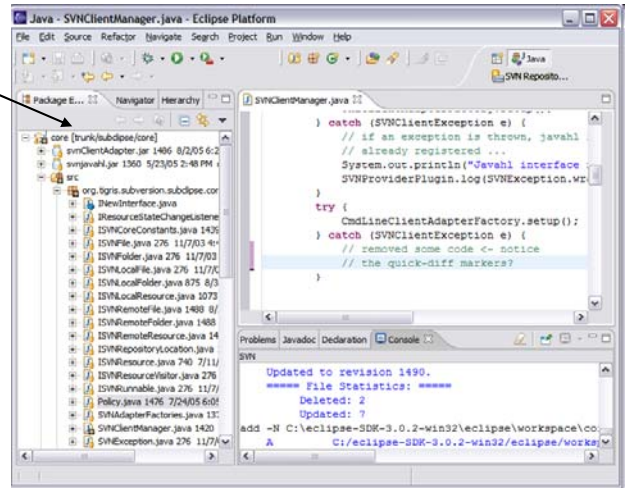


Création de projet

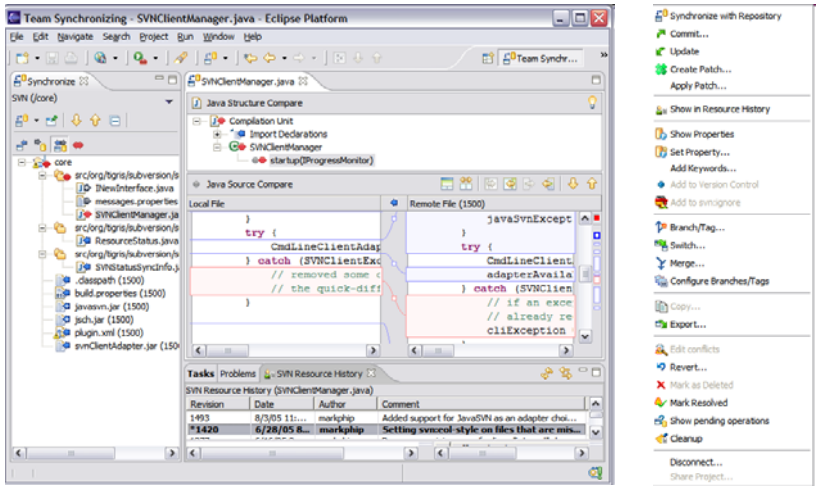


Un projet sous svn dans le workbench

- **Référence au svn**

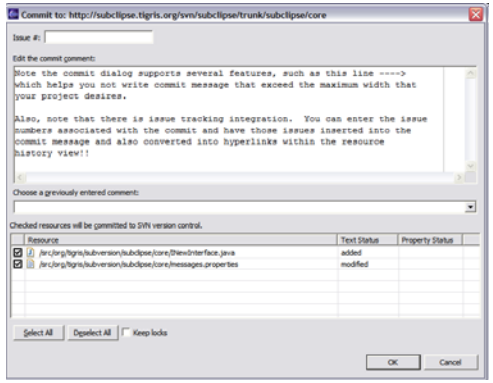


Synchronisation / gestion des conflits



Commit

- **Lors d'un commit, on peut sélectionner**
  - Tout le projet
  - Une sous-partie
  - Un seul fichier
- **On entre un commentaire qui sera stocké dans le svn**
  - (et visible dans le trac pour nous)



Application à votre projet

Université Nice SOPHIA ANTIPOLIS

Eclipse : application à votre projet

- Synchronisation du svn fourni avec subclipse
- Compilation, exécution, débogage dans Eclipse
- Pilotage des tests unitaires pour Java et PHP

Références

- **Site Eclipse**
  - http://www.eclipse.org/
- **Tutoriaux de JM Doudoux (le roi du screenshot !)**
  - http://www.jmdoudoux.fr/accueil_java.htm#dejae
- **Divers supports de cours pour Eclipse (très bien aussi pour les screenshots) :**
  - **http://eclipse.developpez.com/cours/**

Questions