

# Projet de développement

## Introduction à Eclipse

Philippe Collet

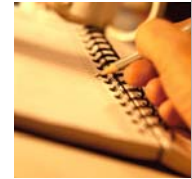


Licence 3 Informatique  
2008-2009



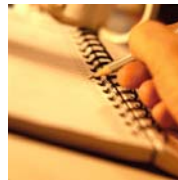
## Organisation

- Cours 1 : principes généraux - svn
- Cours 2 : TRAC 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

- Introduction et historique
- Architecture d'Eclipse
- Démo
- Environnement de développement
- Démo
- JDT
- Démo
- Support CVS / SVN
- Application à votre projet
- Démo...



## 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) : 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 programmeur
- Capable d'automatiser les tâches 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
- ❑ 2008 : Eclipse 3.4 Ganymede

## Sous-projets Eclipse

- ❑ **Eclipse : architecture et structure de la plateforme**
- ❑ **Eclipse Tools : outils pour permettre l'enrichissement 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 (comme le plugin Papyrus utilisé en COO avancée)
- ❑ **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/ProjetDev2009Outils>**
- ❑ **Eclipse**
  - Placé dans /usr/local/eclipse3.4
- ❑ **Java/général**
  - Eclipse 3.4.2 JEE edition : <http://www.eclipse.org/downloads/>
  - plugin subclipse 1.48 (accès au référentiel svn) : <http://subclipse.tigris.org/> (par update)
- ❑ **C**
  - CDT 5.0.2 <http://www.eclipse.org/cdt/downloads.php> (par update)
- ❑ **PHP/MySQL**
  - PDT 2.0.2 : <http://wiki.eclipse.org/PDT/Installation> (par update) : envt standard PHP dans Eclipse
  - SimpleTest Plugin et framework de test : <http://simpletest.org/eclipse/> (par update)
  - SQL Plugin : <http://eclipseql.sourceforge.net/> (par update)

## 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 tâche (pas forcément 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 (dépendances 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

## Plan de travail

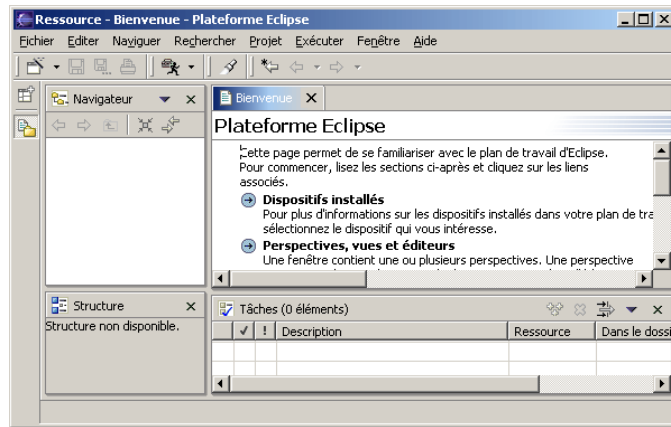
### ❑ Plan de travail = Workbench

- 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

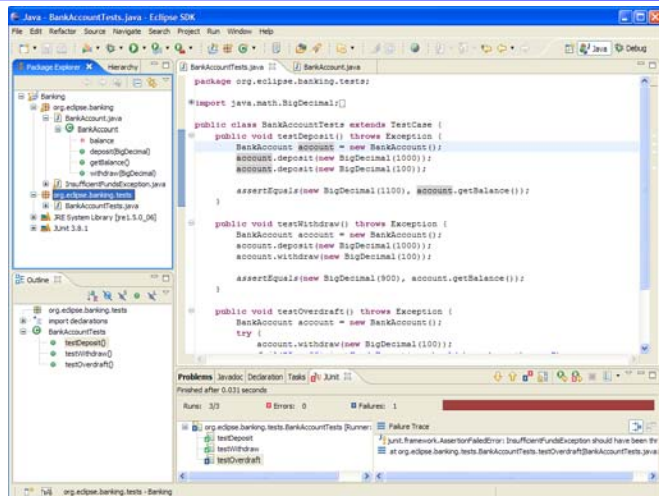
### ☐ Editeur

- Edition plus ou moins dédié (langage avec coloration syntaxique, complétion)
- Exemple : éditeur de texte, éditeur Java

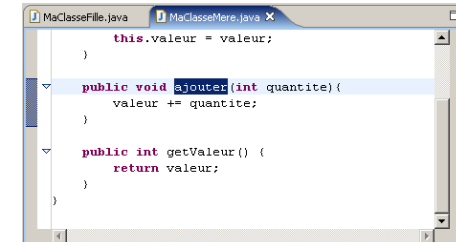
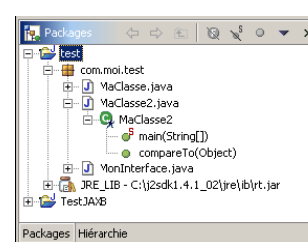
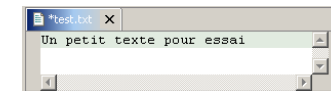
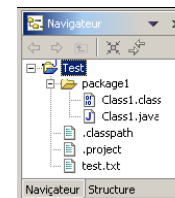
### ☐ Perspective

- ensemble d'éditeurs et vues ayant une disposition précise dans le plan de travail
- Afin de faciliter la réalisation de certaines tâches
- Exemple : navigation, édition Java, exploration de référentiel cvs/svn, synchronisation avec un référentiel
- Le plus étendu des points d'extension

## Perspective



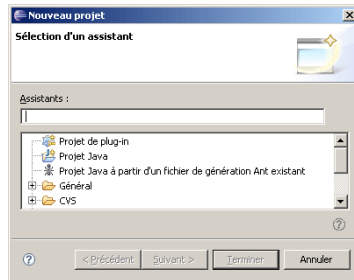
## Vues et éditeurs



## 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)

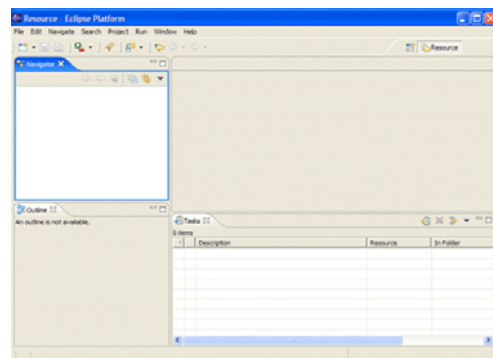
- Exemples : projet Java, site Web.

### □ 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 :

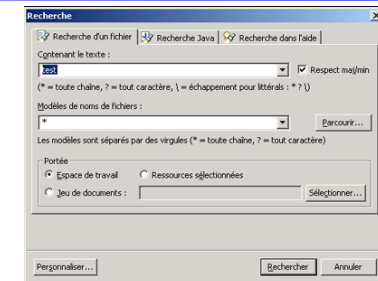
- la vue "Navigateur" qui affiche les ressources (arborescence des fichiers) de l'espace de travail
- un éditeur qui permet d'éditer une ressource sélectionnée dans la vue "Navigateur"
- la vue "Structure" qui permet d'obtenir une arborescence présentant les grandes lignes de certaines ressources en 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

C	Description	Ressource	Dans le dossier	Emplacement
✖	L'importation javax.ejb ne peut pas être résolue	MonPremierEJBBean.java	test_J2EE/com/ma/ejb	ligne 3
✖	SessionBean ne peut pas être résolu ou ne corre...	MonPremierEJBBean.java	test_J2EE/com/ma/ejb	ligne 10 dans ...
✖	Erreur de syntaxe sur le mot clé "else", "case", "d...	TestAssert1.java	test_perso	ligne 21 dans ...

## Aide en ligne

### □ F1 : aide contextuelle

- Dépend de la vue, l'éditeur, etc.

### □ Dans un éditeur :

- CTRL + ESPACE => complétion

# JDT

## JDT : Java Development Tooling

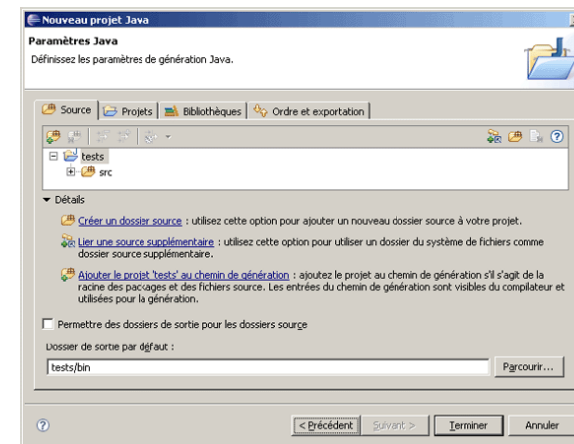
### □ Composants

- les perspectives "Java" et "Navigation Java"
- les vues "Packages" et "Hiérarchie"
- les éditeurs "Java" et "Scrapbook"
- les assistants : pour créer de nouveaux projets, packages, classes, interfaces, ...

### □ Outils de création

- Projet
- Packages
- Classes
- Interfaces

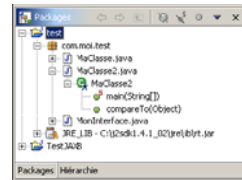
## Projet Java



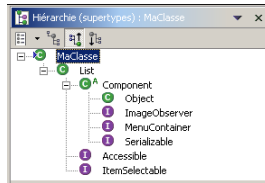
## Vues du JDT

### Package

- Arborescence du code source



### Hiérarchie



### Javadoc, déclarations, erreurs, historique

Ph. Collet

29

## Editeur de code

### Spécifique Java avec les fonctionnalités suivantes

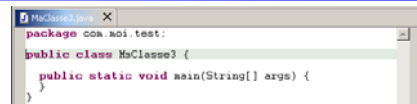
- coloration syntaxique
- complétion de code (CTRL+ESPACE)
- formatage du code source (CTRL+SHIFT+F)
- l'importation et l'exportation de code via un assistant
- forte synergie avec le débogueur

Ph. Collet

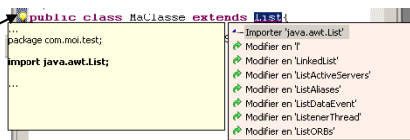
30

## Editeur de code

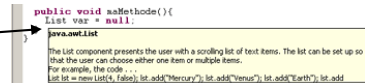
### Coloration syntaxique



### Proposition de correction



### Bulle d'aide

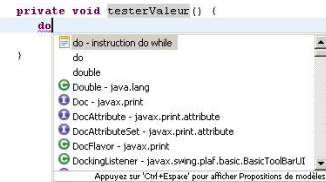
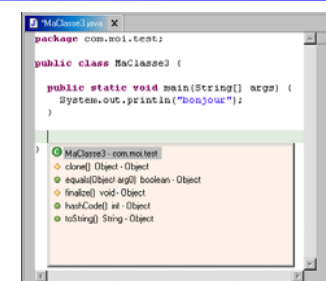
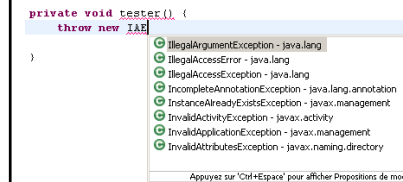
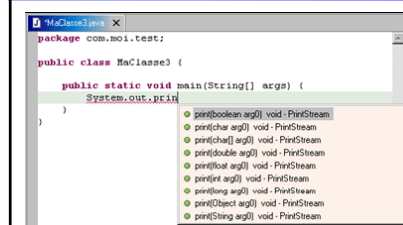


### Formatage, masquage de portion de code, etc.

Ph. Collet

31

## Complétion



Ph. Collet

32

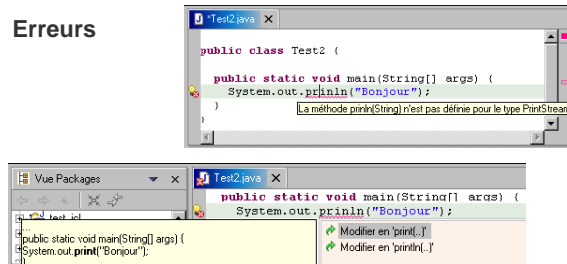


## Compilation et erreur

### Par défaut

- La compilation se fait « tout le temps », en arrière plan

### Erreurs



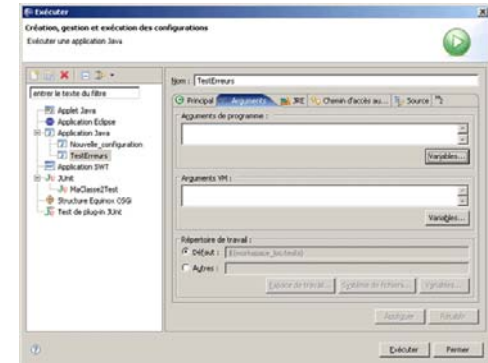
## Exécution

### 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

## Gestion de versions

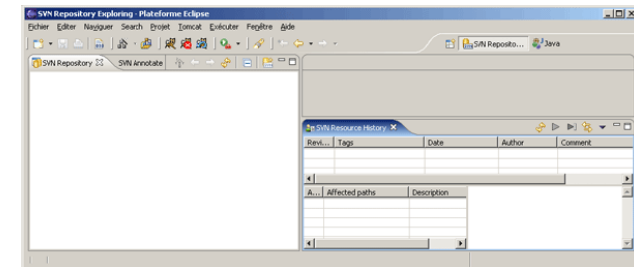
- ❑ **Team support = gestion de versions**
  - Contrôle les versions et le partage d'un projet entre différents développeurs
  - enregistre dans une archive
  - gère des modifications de fichiers
  - récupère toute modification enregistrée
  - visualise les différences entre les versions
- ❑ **CVS (Concurrent 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 (bientôt)**
  - Encore en « incubation »

Ph. Collet

37

## Perspective « SVN Repository Exploring »

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



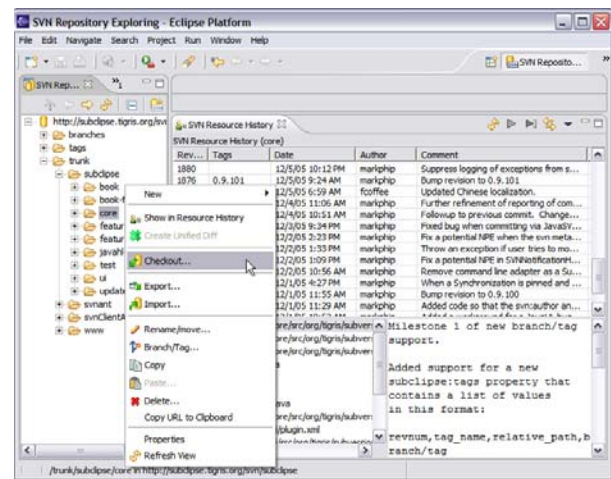
- ❑ **Ajout d'un référentiel**



Ph. Collet

38

## Création de projet

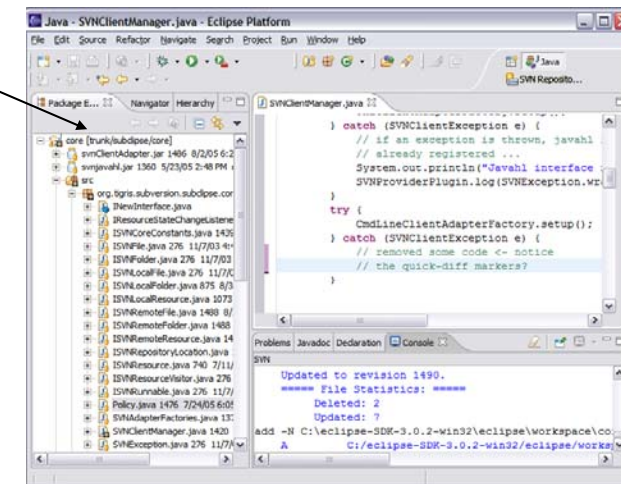


Ph. Collet

39

## Un projet sous svn dans le workbench

- ❑ **Référence au svn**



Ph. Collet

40

## Synchronisation / gestion des conflits

The screenshot shows the Eclipse IDE interface. On the left, the 'Synchronize' view displays the SVN project structure. The main editor shows a 'Java Source Compare' window with two versions of a file side-by-side, highlighting differences in code. The 'Tasks' view at the bottom shows the SVN Resource History for the project.

Ph. Collet 41

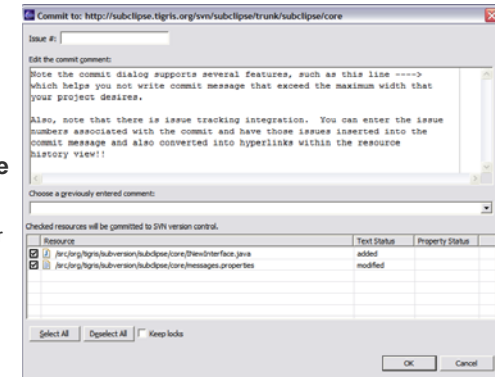
## Commit

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

## 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](http://www.jmdoudoux.fr/accueil_java.htm#dejae)

### ❑ Introduction à Eclipse 3.4 Ganymede

- <http://vogella.developpez.com/tutoriels/eclipse/prise-en-main-eclipse-ganymede/>

## Questions

