

Projet de développement

Introduction à Eclipse

Philippe Collet

Licence 3 MIAGE – S6

2012-2013

http://miageprojet2.unice.fr/index.php?title=User:PhilippeCollet/Projet_de_developpement_2012-2013



Plan

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



Redmine : application à votre projet

□ Par la suite

- Dès les premières séances:
 - ◆ Découpage du travail en composants et milestones (= versions)
 - ◆ 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

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

Eclipse et ses concurrents

❑ Logiciels libres :

- Emacs, XEmacs : basique, mais adaptables à tout langage
- 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 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
- ❑ 2011 : Eclipse 3.7 Indigo
- ❑ 2012 : Eclipse 4.2 Juno

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
- 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.

Installation sur machines fixes du département

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)

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

Ph. Collet

13

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)

Ph. Collet

14

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

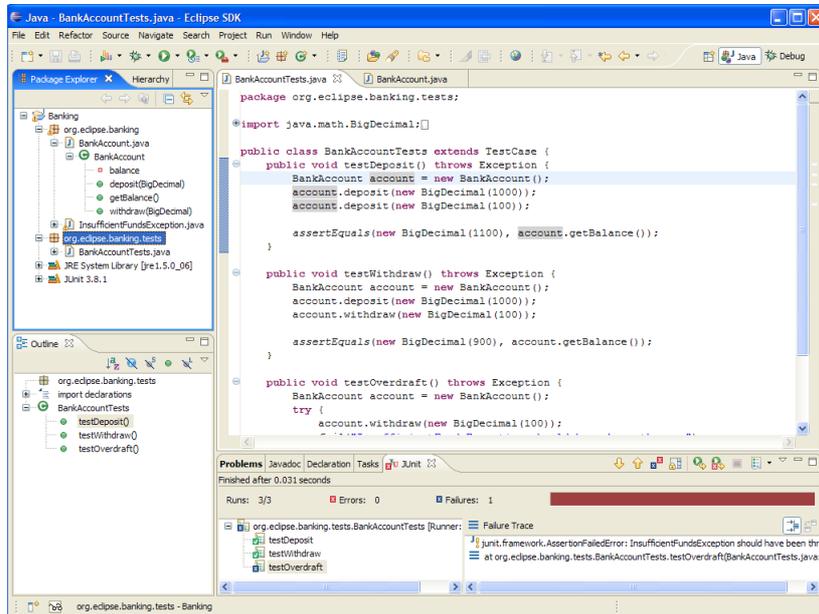
Editeur

- 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

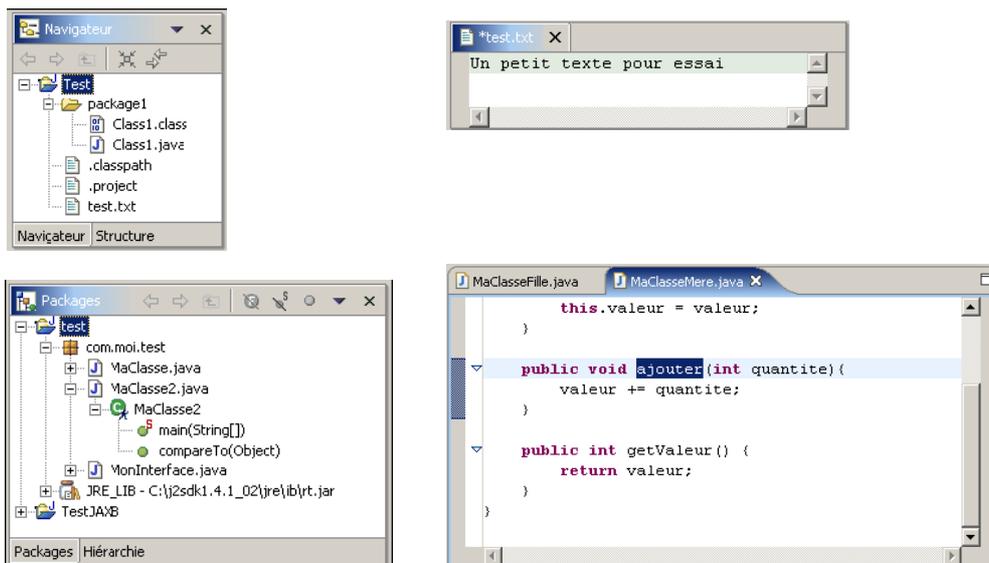
Perspective



Ph. Collet

19

Vues et éditeurs



Ph. Collet

20

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)**

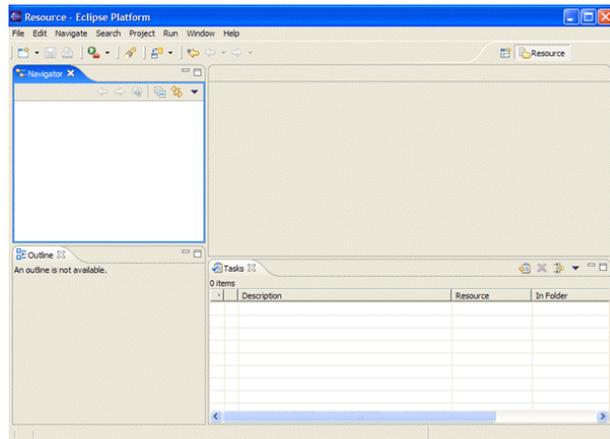
- Exemples : projet Java, site Web.

□ **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



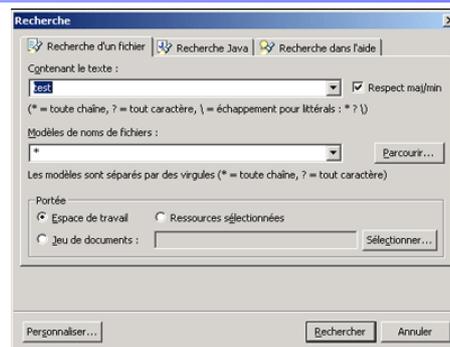
Ph. Collet

23

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

Ph. Collet

24

Aide en ligne

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

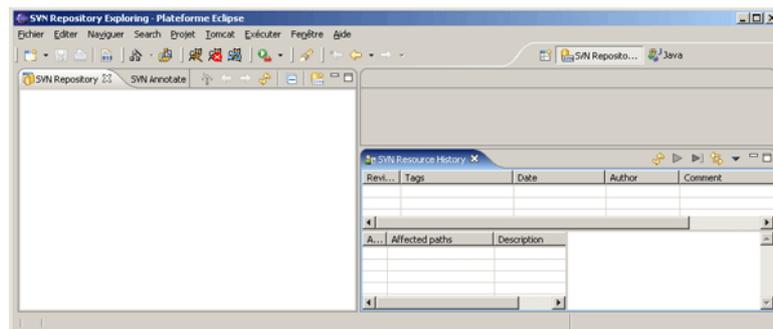
CVS/SVN dans Eclipse

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**
 - Finalement moins stable que Subclipse, à éviter pour l'instant...

Perspective « SVN Repository Exploring »

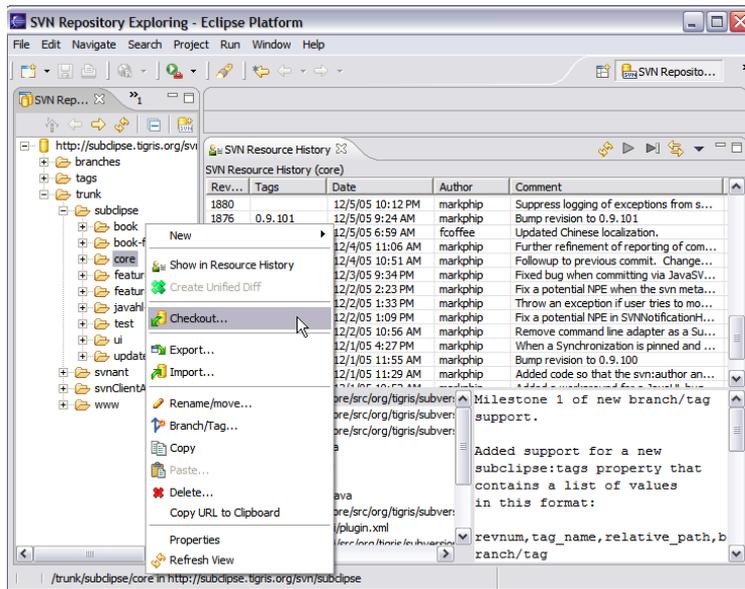
- ❑ **Perspective pour administrer les différents référentiels svn utilisés**



- ❑ **Ajout d'un référentiel**



Création de projet

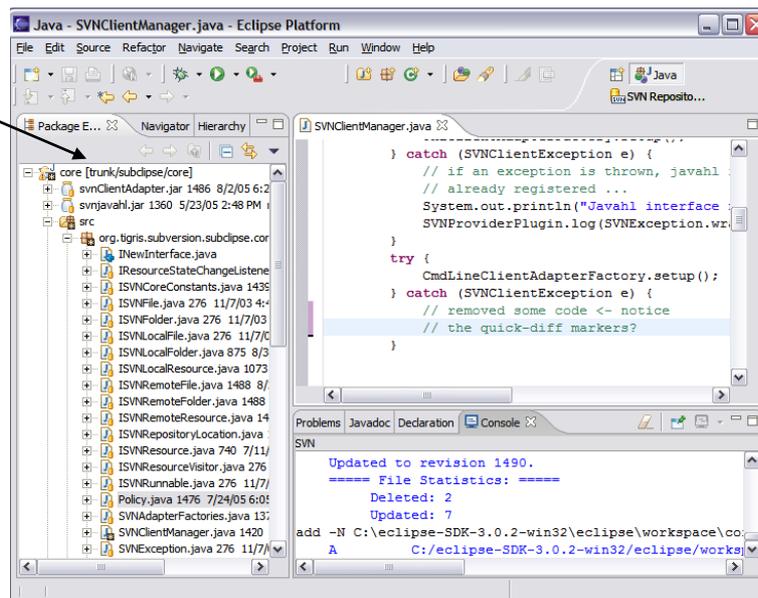


Ph. Collet

29

Un projet sous svn dans le workbench

☐ Référence au svn



Ph. Collet

30

Synchronisation / gestion des conflits

The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Shows the project structure for 'core', including files like 'INewInterface.java', 'SVNClientManager.java', and 'ResourceStatus.java'.
- Source Editor:** Displays a 'Java Source Compare' view for 'SVNClientManager.java', comparing a 'Local File' with a 'Remote File (1500)'. The code shows a try-catch block for 'CmdLineClientAdapter'.
- SVN Resource History:** A table showing the history of changes to 'SVNClientManager.java':

Revision	Date	Author	Comment
1493	8/3/05 11:...	markhip	Added support for JavaSVN as an adapter cho...
1420	6/28/05 8...	markhip	Setting svn:eol-style on files that are mis...
- Right Sidebar:** A list of SVN actions such as 'Synchronize with Repository', 'Commit...', 'Update', 'Create Patch...', 'Apply Patch...', 'Show in Resource History', 'Show Properties', 'Set Property...', 'Add Keywords...', 'Add to Version Control', 'Add to svignore', 'Branch/Tag...', 'Switch...', 'Merge...', 'Configure Branches/Tags', 'Copy...', 'Export...', 'Edit conflicts', 'Revert...', 'Mark as Deleted', 'Mark Resolved', 'Show pending operations', 'Cleanup', 'Disconnect...', and 'Share Project...'.

Ph. Collet 31

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)

The screenshot shows the 'Commit to: http://subclipse.tigris.org/svn/subclipse/trunk/subclipse/core' dialog box. It includes:

- Issue #:** A text input field.
- Edit the commit comment:** A large text area containing instructions and a sample comment: "Note the commit dialog supports several features, such as this line ----> which helps you not write commit message that exceed the maximum width that your project desires. Also, note that there is issue tracking integration. You can enter the issue numbers associated with the commit and have those issues inserted into the commit message and also converted into hyperlinks within the resource history view!!".
- Choose a previously entered comment:** A dropdown menu.
- Checked resources will be committed to SVN version control:** A table listing resources:

Resource	Text Status	Property Status
<input checked="" type="checkbox"/> /src/org/tigris/subversion/subclipse/core/INewInterface.java	added	
<input checked="" type="checkbox"/> /src/org/tigris/subversion/subclipse/core/messages.properties	modified	
- Buttons:** 'Select All', 'Deselect All', 'Keep locks', 'OK', and 'Cancel'.

Ph. Collet 32

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/>