

FAQ Eclipse

Date de publication : 26 août 2008

Dernière mise à jour : 27 décembre 2012

Cette FAQ a été réalisée à partir des questions fréquemment posées sur les forums de www.developpez.net/forums et de l'expérience personnelle des auteurs.

Nous tenons à souligner que cette FAQ ne garantit en aucun cas que les informations qu'elle propose sont correctes. Les auteurs font leur maximum, mais l'erreur est humaine. Cette FAQ ne prétend pas non plus être complète. Si vous trouvez une erreur, ou que vous souhaitez nous aider en devenant rédacteur, lisez Comment participer à cette FAQ ?.

Sur ce, nous vous souhaitons une bonne lecture.

L'équipe Java/Eclipse

Ont contribué à cette FAQ :

L'équipe Eclipse - Lunatix - bobuse - Jfrancois57 -
christopheJ - zork - zsoh - guipom - djo.mos - Vincent
Terret - Laurent.B - Gueritarish - Mickael Baron -

1. Informations (5)	4
2. L'EDI Eclipse (22)	7
3. Développement Java (29)	20
4. Développement J2EE (10)	34
5. Développement J2ME (1)	40
6. La Plate-Forme Eclipse (20)	41
6.1. Développement RCP (2)	43
6.2. Développement de Plug-ins (3)	45
6.3. SWT / JFACE (13)	50
7. Gestion de version (8)	55
8. Plug-ins (16)	60

[Sommaire > Informations](#)**Comment bien utiliser cette FAQ ?****Auteurs : L'équipe Eclipse ,**

Le but : Cette FAQ a été conçue pour être la plus simple possible d'utilisation. Elle tente d'apporter des réponses simples et complètes aux questions auxquelles sont confrontés tous les débutants (et les autres).

L'organisation : Les questions sont organisées par thème, les thèmes pouvant eux-même contenir des sous-thèmes. Lorsqu'une question porte sur plusieurs thèmes, celle-ci est insérée dans chacun des thèmes rendant la recherche plus facile.

Les réponses : Les réponses contiennent des explications et des codes sources. Certaines sont complétées de fichier à télécharger contenant un programme de démonstration. Ces programmes sont volontairement très simples afin qu'il soit aisé de localiser le code intéressant. Les réponses peuvent également être complétées de liens vers d'autres réponses, vers la documentation en ligne de Sun ou vers un autre site en rapport.

Les versions : Lorsqu'une question n'est pertinente que pour certaines version du logiciel, ou lorsque la réponse varie selon la version, une étiquette est placée à gauche de la question. Cette étiquette indique la ou les versions concernées :

Par exemple:

- [3.2] Concerne la version 3.2.
- [3.0-] Concerne les versions 3.0 et antérieures.
- [3.x] Concerne toutes les versions mineures de la version 3.
- [3.4+] Concerne la version 3.4 ainsi que les versions suivantes.

Nouveautés et mises à jour : Lors de l'ajout ou de la modification d'une question/réponse, un indicateur est placé à coté du titre de la question. Cet indicateur reste visible pour une durée de 15 jours afin de vous permettre de voir rapidement les modifications apportées.

J'espère que cette FAQ pourra répondre à vos questions. N'hésitez pas à nous faire part de tous commentaires/remarques/critiques.

lien : [FAQ](#) Comment participer à cette FAQ ?**Remerciements****Auteurs : L'équipe Eclipse ,**

Un grand merci à tous ceux qui ont pris de leur temps pour la réalisation de cette FAQ.

Aux rédacteurs : Remerciements tout d'abord à tous ceux qui ont rédigé les questions et les réponses.

Aux correcteurs : Remerciements également aux personnes qui ont relu les textes pour supprimer un maximum de fautes de français.

Aux visiteurs : Remerciements enfin à tous ceux qui ont consulté cette FAQ, et qui, par leurs remarques, nous ont aidé à la perfectionner.

Et pour finir, un merci tout spécial à tous les membres de l'équipe qui nous ont fourni outils et logiciels nécessaires pour la réalisation de ce document.

Où trouver d'autres sources d'information ?

Auteurs : L'équipe Eclipse ,

Les ressources disponibles sur Developpez.com :

-  [Les cours et tutoriels Eclipse](#)
-  [Les forums Eclipse](#)
-  [La FAQ Eclipse](#)
-  [Les Blogs Eclipse](#)

Les ressources disponibles sur le Site Officiel d'Eclipse :

-  [Eclipse.org](#)
-  [Les plug-ins et applications référencés sur Eclipse Marketplace](#)
-  [Eclipse Live \(Webinaires, Podcasts, Démonos, ...etc\).](#)
-  [Le Bug Tracker des Projets Eclipse](#)
-  [Le blog Planet Eclipse](#)
-  [Le Wiki Eclipse](#)
-  [Le site de téléchargement de la version courante avec toutes ses déclinaisons selon le type de développement que l'on veut faire.](#)

-  [Eclipse Indigo \(3.7\)](#) et  [Les nouveautés significatives](#)
-  [Eclipse Helios \(3.6\)](#) et  [Les nouveautés significatives](#)
-  [Eclipse Galileo \(3.5\)](#) et  [Les nouveautés significatives](#)
-  [Eclipse Ganymede \(3.4\)](#) et  [Les nouveautés significatives](#)
-  [Eclipse Europa \(3.3\)](#)

Comment participer à cette FAQ ?

Auteurs : L'équipe Eclipse ,

Cette FAQ est le fruit d'une collaboration et tout le monde peut y participer. Vous pouvez déposer vos remarques, suggestions ou améliorations [ici](#).

Si vous souhaitez intervenir plus intensément, voici plusieurs compétences actuellement recherchées pour améliorer cette FAQ :

Rédacteur : Bien évidemment, toute nouvelle question/réponse est la bienvenue.

Web designer : Toute personne capable de faire une meilleure mise en page, une feuille de style ou de belles images...

Correcteur : Malgré nos efforts, des fautes d'orthographe ou de grammaire peuvent subsister. Merci de les signaler [ici](#) si vous en débusquez... Idem pour les liens erronés.

lien : [FAQ](#) Quels sont les droits de reproduction de cette FAQ ?

Quels sont les droits de reproduction de cette FAQ ?

Auteurs : L'équipe Eclipse ,

Les codes sources présentés sur cette page sont libres de droits, et vous pouvez les utiliser à votre convenance. Pour le reste, ce document constitue une oeuvre intellectuelle protégée par les droits d'auteurs.

Copyright © 2009 Developpez LLC : Tous droits réservés Developpez LLC. Aucune reproduction ne peut en être faite sans l'autorisation expresse de Developpez LLC. Sinon vous encourez selon la loi jusqu'à 3 ans de prison et jusqu'à 300 000 E de dommages et intérêts. Cette page est déposée à la SACD.

Sommaire > L'EDI Eclipse

Qu'est-ce que Eclipse ?

Auteurs : Lunatix , djo.mos , Laurent.B ,

Eclipse est un EDI (Environnement de Développement Intégré), ou en anglais un IDE (Integrated Development Environment).

Développé par IBM à partir de ses ancêtres Visual Age et Visual Age For Java, il a depuis été rendu open-source et son évolution est maintenant gérée par la  [Fondation Eclipse](#).

Cette fondation est composée d'un grand nombre de  membres et est dirigée par un  conseil d'administration.

Sa conception est complètement modulaire : basée sur des notions telles que "micro noyau  OSGi" (depuis la version 3) ou encore les plug-ins, ce qui fait d'Eclipse une boîte à outils facilement modifiable et extensible.

La license d'Eclipse ( [Eclipse Public License](#) et sa  [FAQ](#)) permet de fournir des plug-ins open sources comme des plug-ins closed-source, des plug-ins gratuits ou encore payants.

C'est pourquoi il est maintenant bien plus qu'un IDE Java, et gère un [FAQ](#) Quels sont les langages que peut gérer Eclipse en plus de Java ? tels que PHP, C, C++, etc.

Eclipse est également devenu une plate-forme, servant de socle à d'autres applications, que celles-ci soient destinées au développement logiciel ou à tout autre domaine.

Cette plate-forme est nommée [FAQ](#) Qu'est ce que Eclipse RCP ?.

En consultant la page  [Eclipse Marketplace](#) vous aurez un aperçu des outils et solutions tournant autour d'Eclipse.

lien : [FAQ](#) Quels sont les langages que peut gérer Eclipse en plus de Java ?

Où trouver Eclipse ?

Auteurs : Lunatix , djo.mos , Laurent.B ,

Sur le site officiel  www.eclipse.org et plus précisément dans la section téléchargements :  [Eclipse downloads](#)

La version officielle en cours à ce jour (28/06/2012) est Eclipse Juno 4.2.

Voici le calendrier des versions simultanées :  http://wiki.eclipse.org/Simultaneous_Release

Eclipse est décliné pour les plates-formes Windows 32/64 bits, Mac 32/64 bits et Linux 32/64 bits.

Eclipse IDE for Java Developers est la version packagée pour faire principalement du développement en langage Java J2SE.

Cependant, d'autres packages sont également disponibles afin de démarrer rapidement dans les domaines ou les langages tels que Java EE, C/C++, Eclipse RCP, Modeling, Reporting, PHP...

Ces packages sont le résultat d'un assemblage cohérent de plug-ins et évitent ainsi de faire soit-même l'ajout des plug-ins nécessaires via l'update manager.

Théoriquement, on peut partir d'une version Classic et la compléter avec les différents plug-ins que l'on juge utiles pour son propre besoin. De ce fait, on peut ajouter sur une même installation d'Eclipse, l'ensemble des plug-ins nécessaires pour faire à la fois du PHP, C/C++, Java/JEE et autres,... Néanmoins, il est préférable de rester raisonnable et de n'installer les plug-ins qu'au fur et à mesure que vos besoins évoluent. Vous éviterez ainsi d'alourdir le fonctionnement d'Eclipse et vous réduirez les conflits potentiels de dépendances entre les versions de plug-ins.

Pour faire votre choix sur un package Eclipse de base, consultez le  [tableau comparatif des différents packages](#).

Vous pouvez trouver les liens de téléchargement des différentes versions d'Eclipse dans ce  [Post-It](#) dans le forum Eclipse.

Où trouver Eclipse en français ?

Auteurs : [Jfrancois57](#) , [djo.mos](#) , [Laurent.B](#) ,

Par défaut, seule la langue anglaise est disponible dans Eclipse. Néanmoins, le projet Babel  <http://babel.eclipse.org/babel/> est une initiative qui tente de permettre, à l'aide de plug-ins supplémentaires, la configuration de l'interface dans différentes langues, notamment le français.

Voici les adresses des update-sites spécifiques aux différentes versions d'Eclipse :

- Juno : <http://download.eclipse.org/technology/babel/update-site/R0.10.0/juno>
- Indigo : <http://download.eclipse.org/technology/babel/update-site/R0.10.0/indigo>
- Helios : <http://download.eclipse.org/technology/babel/update-site/R0.10.0/helios>
- Galileo : <http://download.eclipse.org/technology/babel/update-site/R0.9.1/galileo>
- Ganymede : <http://archive.eclipse.org/technology/babel/update-site/R0.8.1/ganymede>
- Europa : <http://archive.eclipse.org/technology/babel/update-site/R0.8.1/europa>

Attention ! Ces URL ont de fâcheuses tendances à évoluer, donc n'hésitez pas à consulter cette page  <http://www.eclipse.org/babel/downloads.php>.

Et pour les versions archivées, c'est ici  <http://www.eclipse.org/babel/archives.php>.

Si la localisation de votre OS est déjà dans la langue dans laquelle vous souhaitez utiliser Eclipse, il vous suffit en principe de lancer simplement Eclipse et le bon pack de langue sera automatiquement chargé. Si toutefois ce n'était pas le cas, ajoutez simplement le paramètre `-nl fr` dans le raccourci qui vous permet de lancer Eclipse.

Veillez noter ces quelques remarques à propos de Babel :

- Le projet Babel étant géré à part des versions d'Eclipse, il est normal que celui-ci ne soit pas souvent à niveau par rapport à la dernière version en vigueur. Par exemple, au moment de la rédaction de ces lignes, la version officielle en vigueur d'Eclipse est Helios SR2 et donc, la version de Babel n'est adaptée qu'à la version Helios SR1. Pour autant, cela ne doit pas vous empêcher d'installer une version Babel Helios SR1 sur une version d'Eclipse Helios SR2. Simplement, vous noterez sans doute par-ci, par-là quelques libellés en anglais, au lieu d'être en français.
- Le projet Babel a parfois du mal à suivre l'évolution d'Eclipse et cela est sans doute lié au fait qu'il faut du temps et des ressources humaines pour synchroniser toutes les traductions. Donc, pour ceux qui ne peuvent pas s'en passer, soyez heureux que ce projet existe.
- Il arrive que l'installation ne se passe pas toujours sans problème.

Si vous rencontrez des difficultés avec ce plug-in, sans doute mieux vaut-il ne pas trop s'acharner. Le forum est là pour en témoigner.

Dans certains cas, il peut même arriver que des plug-ins ne supportent pas du tout une autre langue que l'anglais, soit parce qu'ils ne sont pas prévus pour être internationalisés, soit qu'ils contiennent des références en dur à des libellés anglais...

- D'autre part, même s'il est plus agréable, car plus simple, d'avoir une interface dans sa langue, gardez à l'esprit qu'il sera bien plus facile d'obtenir de l'aide sur telle ou telle fonctionnalité, à l'aide des termes originaux !

Donc, à moins que l'anglais soit réellement un supplice pour vous, en conservant la version originale, au final vous devriez gagner du temps.

En conséquence, même si le forum developpez.com est un forum francophone, toutes les références à l'IHM d'Eclipse sont données pour les termes anglais uniquement !

lien : [FAQ \[3.3.\] Comment installer un plug-in ?](#)

lien : [FAQ \[3.4.\] Comment installer un plug-in ?](#)

lien : [FAQ \[3.5.\] Comment installer un plug-in ?](#)

Comment forcer Eclipse pour qu'il soit en anglais ?

Auteurs : [Laurent.B](#) ,

Si pour une raison ou pour une autre, vous êtes en face d'une installation d'Eclipse, dont l'interface n'est pas en anglais et que vous préféreriez que ce soit le cas, alors ajoutez simplement -nl en dans le raccourci qui permet de lancer Eclipse.

[2.x] Comment installer Eclipse sur ma machine ?

Auteurs : [Lunatix](#) ,

Décompressez l'archive d'Eclipse. Voilà, c'est fini, Eclipse est installé. Attention il a besoin d'un JRE (Java Runtime Environment) pour fonctionner, voire d'un JDK si vous voulez faire du développement Java.

- L'option -VM permet de spécifier le Runtime Java à utiliser
- Sous Linux, un lien symbolique (ln -s) vers le répertoire jre de Java, dans le répertoire Eclipse, lui permet de trouver l'environnement Java sans plus de configuration

[3.x] Comment installer Eclipse sur ma machine ?

Auteurs : [Lunatix](#) , [Laurent.B](#) ,

Après avoir récupéré un package Eclipse conforme à votre environnement et à vos besoins (cf. [FAQ Où trouver Eclipse ?](#)), décompressez celui-ci et vous pourrez alors lancer Eclipse, en exécutant eclipse.exe sous Windows, ou eclipse sous les autres systèmes.

Toutefois, Eclipse ne pourra démarrer que si une  **Machine Virtuelle Java (d'un JRE ou JDK)** est déjà installée sur votre système, au minimum en version 5.0 ou 6.0 (selon les versions d'Eclipse).

Une fois le JRE ou JDK installé, il se peut qu'Eclipse ne démarre pas pour autant car ne sachant pas où trouver l'un ou l'autre. Il est alors nécessaire que le répertoire bin du JRE/JDK soit référencé dans la variable d'environnement PATH. En tapant java -version en ligne de commande (en vous plaçant par exemple dans

le répertoire d'installation d'Eclipse), vous pourrez voir si tel est bien le cas. Si la commande est inconnue alors il vous faudra paramétrer la variable d'environnement PATH en conséquence.

Néanmoins, le plus simple et le plus pérenne reste de paramétrer Eclipse de façon à ce qu'il fasse usage du JRE/JDK que vous lui aurez indiqué spécifiquement. Ceci est expliqué ici [FAQ Comment paramétrer Eclipse pour qu'il démarre sur une JVM précise ?](#). A noter que la présence de la variable d'environnement JAVA_HOME n'aura pas d'influence sur Eclipse car il ne la prend pas en compte.

 *Si vous comptez développer en Java, vous aurez avantage à faire en sorte qu'Eclipse démarre avec la JVM d'un JDK. Ainsi, vous bénéficierez automatiquement de la Javadoc et du code source de l'API Java Standard. De plus, certaines fonctionnalités, telles que la signature de fichiers jar, requièrent les outils que seul le JDK fournit.*

Si vous comptez utiliser Eclipse pour d'autres usages que du développement en langage Java, ou même si vous êtes encore indécis, alors du moment qu'Eclipse parvient à démarrer à partir d'un JRE, cela est suffisant pour commencer. Vous aurez de toute façon la possibilité de revenir sur cet aspect en temps utile.

 *Sous Windows, il est recommandé de placer Eclipse à un endroit proche de la racine, comme par exemple sous Windows C:\Program Files ou même directement sous C:\ (ou un autre lecteur). Ceci dans le but d'éviter certains problèmes de longueur de chemins, dont les limites peuvent être atteintes assez facilement.*

[2.x] Pourquoi deux versions pour Linux ?

Auteurs : [Lunatic](#) ,

La librairie graphique utilisée par Eclipse (SWT) met en oeuvre l'API graphique du système d'exploitation (mfc sous win32, cocoa sous macos) et Gtk ou Motif sous linux. C'est pourquoi vous trouverez deux versions : une version Eclipse Gtk, et une autre Eclipse Motif.

Pour information, il existe aussi une version QT non disponible à cause de la license de QT (GPL) et il existe également une version non officielle pour Fox  [Eclipse Fox](#)

Qu'est ce que Eclipse RCP ?

Auteurs : [Lunatic](#) ,

RCP : Rich Client Platform

Il s'agit d'un framework de développement pour client riche sur lequel est basé Eclipse. En effet même si Eclipse a été principalement créé pour être un logiciel de développement, sa nature modulaire le rend utilisable pour n'importe quel type d'application.

Eclipse RCP est composé d'un conteneur OSGi (Equinox) ainsi que d'un ensemble minimal de plug-ins pour faire fonctionner une application graphique (environ 13 Mo).

La page du projet  wiki.eclipse.org/index.php/Rich_Client_Platform

Le site de news liés à Eclipse  PlanetEclipse.org

Des applications basées sur le RCP  eclipse.org/community/rcp.php

Notons enfin que comme pour SWT, il est possible de faire un projet commercial en utilisant RCP sans rien avoir à payer.

Comment paramétrer Eclipse pour qu'il démarre sur une JVM précise ?

Auteurs : L'équipe Eclipse , Laurent.B ,

Si un JRE ou un JDK est installé sur votre système, alors Eclipse peut théoriquement démarrer sans problème.

Si vous avez plusieurs JRE/JDK installés sur votre système et que vous souhaitez qu'Eclipse démarre sur l'un d'eux en particulier, alors différentes possibilités s'offrent à vous :

- 1 Il est possible de spécifier la JVM (machine virtuelle Java) à l'aide de l'argument `-vm`, suivi du chemin vers l'exécutable Java.

Exemple sous Windows :

```
eclipse.exe -vm "C:\Program Files\Java\jdk1.6.0\bin\javaw.exe"
```

A vous d'adapter en fonction de votre OS et de l'emplacement de l'exécutable Java.

Ceci fonctionne à la fois en ligne de commande ou à l'aide de l'icône qui sert à lancer Eclipse.

- 2 L'argument `-vm` peut également être spécifié dans le fichier `eclipse.ini`. Il doit cependant être indiqué de cette manière :

```
...  
-vm  
C:/Program Files/Java/jdk1.6.0/bin/javaw.exe  
...
```

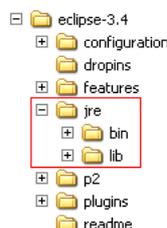
Cette déclaration peut être placée au début du fichier, ou plus bas, cependant elle ne doit jamais couper les paramètres et leurs valeurs et de plus, elle doit toujours être ajoutée avant le paramètre `-vmargs`, sans quoi vous obtiendrez une erreur expliquant qu'Eclipse ne peut pas démarrer.

Ce fichier n'étant pas particulièrement intuitif, le mieux est de placer le paramètre `-vm` et le chemin qui l'accompagne, juste avant le paramètre `-vmargs`. Donc, comme sur l'exemple suivant :

```
...  
-vm  
D:\Program Files\Java\jdk1.6.0_35\bin\javaw.exe  
-vmargs  
-Dosgi.requiredJavaVersion=1.5  
-Xms128m  
-Xmx512m
```

- 3 Sous Windows, une autre solution consiste à copier le contenu d'un JRE qui aura préalablement été installé de façon classique, dans un répertoire nommé `jre` et cela, directement dans le répertoire d'installation d'Eclipse.

Concrètement, l'arborescence doit être proche de ceci :



Sous Linux, le même principe peut être réalisé avec un lien symbolique (ln -s) vers le répertoire jre de Java, donc sans copier quoi que ce soit.

Eclipse n'est alors plus dépendant des installations, désinstallations ou mises à jour de JDK / JRE. Ainsi, lors du démarrage, si rien n'est spécifié dans le fichier eclipse.ini ou dans l'icône de lancement, alors le répertoire jre sera automatiquement pris en compte.

Pour vérifier si tout est cohérent, une fois Eclipse démarré, il suffit d'aller dans le menu Help > About Eclipse Platform > Configuration Details et de chercher le paramètre "eclipse.vm=..." afin de vérifier que le chemin indiqué est bien celui auquel on s'attend.

Comme expliqué ci-dessus, il y a différents moyens d'indiquer à Eclipse, quelle JVM utiliser pour son exécution. Ainsi, il procèdera par élimination, donc en cherchant d'abord si l'argument -vm est précisé en paramètre d'exécution ou dans le fichier eclipse.ini. A défaut, il vérifiera s'il n'y a pas un répertoire jre dans son propre répertoire d'installation. Et pour finir, s'il n'y a rien de tout cela, il tentera de chercher par rapport à la variable d'environnement PATH, ou finalement de s'adresser au système pour déterminer si une JVM est disponible quelque part. En cas d'échec de toutes ces possibilités, il doit en principe indiquer clairement qu'il n'a pas trouvé de JVM.

Comment ajuster la mémoire utilisée par Eclipse pour sa propre exécution ?

Auteurs : **Laurent.B** ,

Lorsque l'on installe Eclipse, celui-ci est pré-configuré avec une certaine quantité de mémoire dont il peut disposer pour son exécution.

Comme cet IDE est basé sur Java, il a naturellement hérité des paramètres standards pour l'allocation de mémoire d'une application Java, c'est-à-dire Xmx et Xms.

De fait, il est notamment possible de configurer ces deux paramètres au sein du fichier eclipse.ini (présent dans le répertoire racine de votre installation d'Eclipse).

Le plus important est le paramètre de la mémoire Xmx. Donc, lorsque cela s'avère nécessaire (si par exemple Eclipse commence à indiquer qu'il n'a plus assez de mémoire), vérifiez que le paramètre Xmx n'est pas plafonné à 256m ou 512m, car ce sont des quantités de mémoire relativement basses (mais néanmoins suffisantes dans bien des cas).

Par exemple, mettez la valeur suivante :

```
-Xmx768m
```

ou davantage si nécessaire

```
-Xmx1024m
```

En respectant bien le formalisme et également en plaçant ce paramètre après -vmargs.

La mémoire Xms est la mémoire initiale. Si vous avez suffisamment de mémoire pour allouer la totalité de la mémoire Xmx, alors vous pouvez mettre une valeur identique pour Xmx et Xms. Procédez comme pour la mémoire Xmx, indiquez la valeur qui vous convient.

Par exemple comme ceci :

```
-Xms768m
```

En revanche, vous ne pourrez pas avoir une valeur Xms supérieure à celle de Xmx.

Si vous n'avez pas assez de mémoire vive, voire si vous avez atteint un maximum (en fonction de l'OS), Eclipse ne démarrera pas et affichera une fenêtre de rapport d'erreur. A ce moment-là, réduisez la quantité de mémoire Xmx, jusqu'à ce que ça passe. Vous avez la possibilité de mettre des valeurs arbitraires, il n'est pas utile de respecter les exposants de 2.

A noter que vous pouvez utiliser ces mêmes paramètres via la ligne de commande ou dans un raccourci, par exemple de cette manière :

```
eclipse.exe -vmargs -Xmx400m -Xms400m
```

Vous pouvez contrôler votre paramétrage, une fois Eclipse démarré, en affichant le moniteur de mémoire. Cette opération est expliquée dans cette [FAQ](#) Comment récupérer de la mémoire lorsqu'Eclipse arrive à la limite de sa mémoire allouée ?

Comment récupérer de la mémoire lorsqu'Eclipse arrive à la limite de sa mémoire allouée ?

Auteurs : [Laurent.B](#) ,

Depuis bien des versions, Eclipse sait nous prévenir, théoriquement, lorsque la limite de sa mémoire allouée est bientôt atteinte. Ceci a le grand avantage d'éviter de perdre ce qui n'a pas encore été sauvegardé (des suites d'un plantage sec), voire de libérer de la mémoire afin de revenir à un niveau moins critique.

Pour libérer la mémoire, il y a une première astuce simple qui consiste à fermer des vues, éditeurs ou perspectives. Cependant, en procédant ainsi, on peut également perdre un certain contexte et donc cela peut être potentiellement gênant.

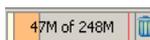
Intervient alors la possibilité de provoquer le Garbage Collector (ramasse-miettes), lequel va bien souvent pouvoir nous dépanner, en évitant d'avoir à redémarrer Eclipse.

Pour exécuter le Garbage Collector, il faut préalablement avoir activé le moniteur de mémoire, via le menu **Window > Preferences > General**, puis en cochant la case **Show heap status**.

Une fois celle-ci activée et après avoir cliqué sur OK, on peut voir apparaître (en bas à droite en principe), le moniteur de mémoire. Il ne reste plus ensuite qu'à cliquer sur le petit icône de la poubelle, à droite de celui-ci.

On peut déclencher plusieurs fois le Garbage Collector car parfois cela permet de libérer davantage de mémoire. Cependant, il se peut également que la mémoire ne désemplisse pas et il faut alors tout de même penser à fermer un éditeur ou toute vue ou élément dont on sait, ou l'on devine, qu'il va pouvoir restituer de la mémoire. Quitte à le rouvrir juste après.

Voici à quoi ressemble le moniteur de mémoire



Vous constaterez que la mémoire totale affichée est toujours inférieure à la mémoire paramétrée (sic).

Lorsque la mémoire arrive à saturation, vous pourrez constater que la zone devient rouge. Sur certaines opérations, il arrive que le seuil maximum soit atteint, sans pour autant provoquer de problème. Une fois

l'opération gourmande en mémoire, terminée, la zone redevient orange. Si ce n'est pas le cas, alors aidez-vous du Garbage Collector pour faire redescendre le niveau.

Qu'est-ce qu'un workspace ?

Auteurs : Lunatix , djo.mos ,

Il s'agit de l'espace de travail : quoi vous aviez deviné ? Plus précisément, c'est un répertoire dans lequel vous allez pouvoir créer vos projets, qu'ils soient Java ou autres. Tous les fichiers manipulés par Eclipse se trouvent dans ce répertoire. Bien sûr, vous pouvez disposer de plusieurs workspaces, dans lesquels vous pourrez regrouper vos projets comme vous le souhaitez.

- Avoir trop de projets ouverts peut ralentir Eclipse
- Copier/Coller un workspace suffit généralement à le déplacer. Ceci autorise donc la sauvegarde, le déplacement d'un poste à un autre, d'emporter son boulot sur une clé USB, ...

Notez que Eclipse et plusieurs plug-ins utilisent le workspace pour y stocker des paramètres de configuration dans un sous-dossier caché .metadata.

Ainsi, effacer ce dossier ou démarrer Eclipse avec l'option -clean va supprimer ces paramètres.

Comment définir un répertoire différent pour mon workspace ?

Auteurs : L'équipe Eclipse ,

Depuis la version 3.0, Eclipse demande à chaque démarrage quel répertoire utiliser en tant que Workspace. Une fois le démarrage terminé, vous avez la possibilité de basculer sur un autre Workspace en allant dans le menu File > Switch workspace.

Vous pouvez aussi, depuis Eclipse 2, spécifier un répertoire particulier par l'intermédiaire de l'option -data. Celle-ci peut être spécifiée en ligne de commande de la manière suivante : eclipse -data "c:\dev\ eclipse-workspace-2"

Vous pouvez également paramétrer un icône de manière à ce qu'il ouvre Eclipse avec le bon Workspace. Le principe est exactement le même qu'en ligne de commande et il suffit de préciser le paramètre -data suivi du répertoire workspace souhaité, au niveau de la cible de lancement.

Qu'est ce qu'une Perspective, une View ou un Editor ?

Auteurs : Lunatix , Laurent.B ,

Contrairement à un grand nombre d'applications, Eclipse n'est pas basé sur MDI (Multi Document Interface), mais sur un système différent qui lui est propre : les Perspectives.

Une perspective est un ensemble de vues qui permet de travailler sur un thème donné.

On a par exemple la perspective Resource qui permet de voir tout ce que contient le workspace, la perspective Java pour développer en Java, la perspective CVS pour gérer ses repositories, la perspective Debug pour le débogage, ...

Ces perspectives sont accessibles via la petite barre d'outils qui se trouve en haut à droite (par défaut)



Les vues (ou views) sont des fenêtres graphiques d'Eclipse qui permettent d'afficher des informations : une vue "Package Explorer" va permettre de voir les packages Java du projet, une vue CVS History pour afficher l'historique d'un fichier sous CVS, ...etc.

La plupart des plug-ins ajoutent des perspectives (par exemple une perspective base de données pour QuantumDB) composées de vues (Table Explorer) Les plug-ins les plus petits n'ajoutent parfois que des vues comme le plug-in TreeHistory qui ajoute une vue graphique (à la WinCVS) pour CVS.

Il est possible de customiser les perspectives : pour ouvrir par exemple une vue CVS repositories dans la perspective Java, il suffit d'ouvrir la perspective Java et de faire : Window > Show View et sélectionner la vue que l'on veut en plus. Cette modification est permanente. Cela vous permet donc de customiser les perspectives que vous utilisez et de conserver ces paramètres.

Reset Perspective : vous êtes allé trop loin, la vue Ruby ouverte dans la perspective Java entre la vue CVS et la vue Table Explorer de Quantum DB, ce n'est plus possible. Window > Reset Perspective... et voilà la perspective Java comme au premier jour.

[2.x] Comment sauvegarder ses préférences ?

Auteurs : L'équipe Eclipse ,

Allez dans le menu Window > Preferences, cliquez sur le bouton Export (en bas à gauche), puis choisissez ce que vous voulez sauvegarder.

Pour les ré-importer, allez dans le menu Window > Preferences, cliquez sur le bouton Import (en bas à gauche)

[3.x] Comment sauvegarder ses préférences ?

Auteurs : Laurent.B ,

Eclipse permet d'exporter ses préférences dans un fichier et de les re-importer ensuite dans une autre installation d'Eclipse.

Pour exporter les préférences dans un fichier, procédez comme suit :

- 1 Allez dans le menu File > Export.
- 2 Sélectionnez General > Preferences puis cliquez sur Next.
- 3 Sélectionnez les éléments qui vous intéressent.
- 4 Indiquez un emplacement et un nom de fichier d'export en cliquant sur Browse....
- 5 Terminez en cliquant sur Finish

Pour importer les préférences à partir d'un fichier, procédez comme suit :

- 1 Allez dans le menu File > Import.
- 2 Sélectionnez General > Preferences et cliquez sur Next.
- 3 Cliquez sur Browse... et indiquez l'emplacement du fichier à importer.
- 4 Sélectionnez les éléments qui vous intéressent.

5 Cliquez sur Finish.

lien : [FAQ Comment sauvegarder ses configurations d'exécution et ses points d'arrêt de debug ?](#)

Comment sauvegarder ses configurations d'exécution et ses points d'arrêt de debug ?

Auteurs : [Laurent.B](#) ,

Pour sauvegarder les configurations de lancement de vos projets, procédez comme suit :

- Allez dans le menu File > Export... > Run/Debug
- Sélectionnez Launch Configurations puis cliquez sur Next
- Renseignez l'emplacement du répertoire où vont être créés vos fichiers de configuration (1 fichier .launch par projet).
- Cliquez sur Finish

Pour importer des configurations de lancement, procédez comme suit :

- Allez dans le menu File > Export... > Run/Debug
- Sélectionnez Launch Configurations puis cliquez sur Next
- Renseignez l'emplacement du répertoire où sont présents vos fichiers de configuration de lancement (dont l'extension est .launch).
- Sélectionnez les fichiers qui vous intéressent, puis cliquez sur Finish

Le principe est identique pour l'import et l'export de Points d'Arrêt (Breakpoints). Les fichiers de Breakpoints comportent quant à eux l'extension .bkpt.

Comment utiliser Ant avec Eclipse ?

Auteurs : [christopheJ](#) , [Laurent.B](#) ,

Eclipse intègre par défaut un plug-in Ant.

Pour être sûr qu'il est bien installé, si vous n'avez pas de fichier nommé build.xml déjà présent dans un projet, créez-en un et ouvrez-le.

L'ouverture de ce fichier entraîne en principe l'activation d'un éditeur XML spécifique à Ant, lequel permet notamment l'auto-complétion relative à ce dernier.

Pour exécuter le fichier build.xml avec Ant, il suffit de faire un clic droit sur le fichier, puis de sélectionner le menu Run.

Il est également possible, après avoir sélectionné le fichier, de passer par le menu Run > External Tools > Run as > Ant Build.

D'autre part, si vous avez la vue Ant ouverte, faites glisser votre fichier à l'intérieur afin que celui-ci soit pris en compte. Vous pouvez ainsi exécuter très facilement des tâches particulières, si ce n'est pas la tâche par défaut dont vous avez besoin.

Si votre vue Ant n'est pas ouverte, alors ouvrez-la à l'aide du menu Window > Show View > Other... > Ant > Ant.

Où trouver l'aide d'Eclipse ?

Auteurs : Jfrancois57 , Laurent.B ,

Eclipse intègre théoriquement sa propre documentation. Elle est accessible par le menu : Help > Help Contents.

D'autre part, il existe une documentation accessible en ligne pour les versions suivantes :

- Juno (4.2) :  <http://help.eclipse.org/juno/>
- Indigo (3.7) :  <http://help.eclipse.org/indigo/>
- Helios (3.6) :  <http://help.eclipse.org/helios/>
- Galileo (3.5) :  <http://help.eclipse.org/galileo/>
- Ganymede (3.4) :  <http://help.eclipse.org/ganymede/>
- Europa (3.3) :  <http://help.eclipse.org/help33/>

Quelles sont les options de lancement de Eclipse ?

Auteurs : Laurent.B ,

Eclipse autorise un certain nombre d'options destinées à paramétrer son lancement.

La liste est assez longue et permet de préciser par exemple, les arguments de la machine virtuelle, le workspace cible, l'activation du mode debug, le nettoyage du cache, la localisation et bien d'autres.

La liste des options de lancement d'Eclipse est décrite dans la documentation, accessible par le menu Help > Help Contents, puis la rubrique Workbench User Guide > Tasks > Running Eclipse.

Comment basculer en mode sélection de bloc ?

Auteurs : Laurent.B ,

L'éditeur d'Eclipse propose un mode de sélection/édition en forme de bloc rectangulaire, qui peut s'avérer très utile dans certaines circonstances.

Comme c'est un mode de fonctionnement, il faut l'activer/désactiver à l'aide du raccourci ALT + SHIFT + A. Ainsi, le curseur est transformé en croix et autorise la sélection de manière graphique, d'une zone de texte.

L'utilité principale de cette fonctionnalité est qu'il est alors possible de modifier en colonne, le texte sélectionné. Cela peut remplacer, de façon assez appréciable un "rechercher/remplacer".

Une autre utilisation, est de facilement récupérer un bout de code, sans récupérer tous les caractères présents en amont, afin par exemple de le copier/coller dans un post, sur le forum developpez.com.

Voici un aperçu de la façon dont ça se présente :

Sélection

```

14 private static void roundValueToTwoDecimals()
15 {
16     int i = 1;
17     System.out.println(i++ + " : " + Math.round(0.00 * 100f) / 100f);
18     System.out.println(i++ + " : " + Math.round(0.01 * 100f) / 100f);
19     System.out.println(i++ + " : " + Math.round(0.051 * 100f) / 100f);
20     System.out.println(i++ + " : " + Math.round(0.054 * 100f) / 100f);
21     System.out.println(i++ + " : " + Math.round(0.055 * 100f) / 100f);
22     System.out.println(i++ + " : " + Math.round(1.01111 * 100f) / 100f);
23     System.out.println(i++ + " : " + Math.round(1.04111 * 100f) / 100f);
24     System.out.println(i++ + " : " + Math.round(1.05111 * 100f) / 100f);
25     System.out.println(i++ + " : " + Math.round(1.054111 * 100f) / 100f);
26     // 10
27     System.out.println(i++ + " : " + Math.round(1.05499 * 100f) / 100f);
28
29     System.out.println(i++ + " : " + Math.round(1.055111 * 100f) / 100f);
30     System.out.println(i++ + " : " + Math.round(1.056111 * 100f) / 100f);

```

Modification

```

14 private static void roundValueToTwoDecimals()
15 {
16     int i = 1;
17     log.debug(i++ + " : " + Math.round(0.00 * 100f) / 100f);
18     log.debug(i++ + " : " + Math.round(0.01 * 100f) / 100f);
19     log.debug(i++ + " : " + Math.round(0.051 * 100f) / 100f);
20     log.debug(i++ + " : " + Math.round(0.054 * 100f) / 100f);
21     log.debug(i++ + " : " + Math.round(0.055 * 100f) / 100f);
22     log.debug(i++ + " : " + Math.round(1.01111 * 100f) / 100f);
23     log.debug(i++ + " : " + Math.round(1.04111 * 100f) / 100f);
24     log.debug(i++ + " : " + Math.round(1.05111 * 100f) / 100f);
25     log.debug(i++ + " : " + Math.round(1.054111 * 100f) / 100f);
26     // 10
27     System.out.println(i++ + " : " + Math.round(1.05499 * 100f) / 100f);
28
29     System.out.println(i++ + " : " + Math.round(1.055111 * 100f) / 100f);
30     System.out.println(i++ + " : " + Math.round(1.056111 * 100f) / 100f);

```

Les raccourcis claviers indispensables

Auteurs : L'équipe Eclipse , Laurent.B ,

Quelques raccourcis clavier indispensables sous Eclipse :

Raccourci	Description
CTRL + SHIFT + L	Affiche la liste des raccourcis (sans doute le plus important à retenir)
CTRL + Espace	L'indispensable auto-complétion
CTRL + SHIFT + R	Recherche d'un fichier dans le workspace
CTRL + SHIFT + F	Mise en forme du code (vous pouvez surligner une zone de code pour restreindre le formatage)
CTRL + SHIFT + I	Indentation du code (vous pouvez surligner une zone de code pour restreindre l'indentation)
CTRL + D	Efface la ligne courante
CTRL + SHIFT + C	Commente / Décommente les lignes sélectionnées (commentaire encapsulant chaque ligne indépendamment)
CTRL + SHIFT + /	Commente / Décommente un blocs de lignes sélectionnées (peut avoir le

	même comportement que CTRL + SHIFT + C selon le contexte)
CTRL + SHIFT + P	Pour se déplacer d'une accolade à l'autre
CTRL + K et CTRL + SHIFT + K	Positionne le curseur sur l'occurrence suivante ou précédente de la sélection de départ (attention réagit en fonction du paramétrage courant de la recherche avec CTRL + F)
CTRL + ALT + Flèche haut et CTRL + ALT + Flèche bas	Duplique la sélection courante ou la ligne courante au-dessus ou en-dessous.
ALT + Flèche haut et ALT + Flèche bas	Décale d'une ligne la sélection courante ou la ligne courant, au-dessus ou en-dessous.

Evidemment, il y en a plein d'autres mais mieux vaut-il retenir au moins CTRL + SHIFT + L, car on a vite fait de les oublier.

Parfois, certains raccourcis sont absents. Alors il est possible de configurer soi-même sa combinaison en allant dans le menu Window > Preferences > General > Keys, ou en faisant CTRL + SHIFT + L + L (deux fois L).

lien : [FAQ](#) Les raccourcis claviers indispensables pour la programmation Java

Sommaire > Développement Java

Les raccourcis claviers indispensables pour la programmation Java

Auteurs : L'équipe Eclipse , Laurent.B ,

Quelques raccourcis clavier indispensables pour la programmation Java sous Eclipse :

Raccourci	Description
CTRL + Espace	L'indispensable auto-complétion
CTRL + SHIFT + O	Organise les imports
CTRL + SHIFT + T	Recherche une classe dans le workspace (indispensable s'il on a beaucoup de fichiers)
CTRL + O	Affichage les attributs et méthodes de la classe courante
CTRL + O + O ('O' deux fois)	Ajoute à l'affichage les attributs et méthodes héritées
CTRL + T	Affiche l'arborescence d'héritage de la classe courante
ALT + SHIFT + J	Génère un template de Javadoc pour une classe une méthode ou un attribut en fonction de la sélection
CTRL + SHIFT + F	Mise en forme du code (vous pouvez surligner une zone de code pour restreindre le formatage)
ALT + SHIFT + R	Renommer une variable, une méthode ou fonction, une classe... etc (depuis 3.3 la transformation peut se faire en temps réel)
CTRL + SHIFT + C	Commente / Décommente les lignes sélectionnées (commentaire encapsulant chaque ligne indépendamment)
CTRL + SHIFT + /	Commente / Décommente un blocs de lignes sélectionnées (peut avoir le même comportement que CTRL + SHIFT + C selon le contexte)
CTRL + SHIFT + P	Pour se déplacer d'une accolade à l'autre

Parfois, certains raccourcis sont absents. Alors il est possible de configurer soi-même sa combinaison en allant dans le menu Window > Preferences > General > Keys, ou en faisant CTRL + SHIFT + L + L (deux fois L).

lien : [FAQ Les raccourcis claviers indispensables](#)

Comment créer un projet Java ?

Auteurs : djo.mos ,

Pour créer un projet Java, allez dans le menu File > New > Project... et localisez Java Project.

Dans l'assistant qui s'affiche, saisissez un nom pour le projet à créer et cliquez sur Finish.

Qu'est-ce que le Build Path d'un projet Java ?

Auteurs : Laurent.B ,

Le Build Path est ce qui constitue le référencement des fichiers, répertoires, bibliothèques (jar), c'est-à-dire toute ressource nécessaire à la compilation d'un programme Java au sein d'Eclipse. Cet élément correspond de fait à l'option -classpath utilisée avec la commande javac, en ligne de commande.

Sous Eclipse, chaque projet Java possède un Build Path que l'on peut ajuster selon ses besoins. Il se configure via un clic droit sur le projet, puis Properties > Java Build Path. Sur cette page de préférences, on trouve notamment les onglets Source, Projects et Libraries, lesquels permettent de référencer différents types de ressources.

Au minimum un projet Java doit avoir un répertoire pour les fichiers source, lequel doit être défini dans l'onglet Source mais aussi, un JRE System Library défini dans l'onglet Libraries.

Bien que de base, un projet Java puisse se contenter de l'API standard de Java (fournie par le JRE ou le JDK) pour compiler et s'exécuter, il est rare que cela soit suffisant. De fait, vous pouvez agrémenter votre Build Path avec entre autres, des bibliothèques tierces.

L'ajout de ces bibliothèques est décrit ici Comment ajouter un JAR à mon projet ?.

Remarque : Dans Eclipse, il ne faut pas confondre Build Path et Class Path. Le Class Path concerne uniquement l'exécution d'un programme Java. Le principe est similaire à celui du Build Path et d'ailleurs ce dernier sert de base pour la constitution du Class Path mais le Class Path est propre à l'exécution et peut comporter d'autres ressources qui ne sont pas nécessaires lors de la compilation. Cf. Qu'est-ce que le Class Path d'un projet Java ?

Qu'est-ce que le Class Path d'un projet Java ?

Auteurs : Laurent.B ,

Le Class Path est ce qui constitue le référencement des fichiers, répertoires, bibliothèques (jar), c'est-à-dire toute ressource nécessaire à l'exécution d'un programme Java au sein d'Eclipse. Il correspond bien évidemment au paramètre -classpath que l'on peut utiliser lors de l'appel des programmes Java, en ligne de commande.

Lors d'une première exécution de votre application Java, un Class Path est déterminé automatiquement. On peut toutefois le constituer spécifiquement et manuellement, à partir du menu Run > Run Configurations..., puis en ajoutant une configuration d'exécution de type Java Application (par exemple). Ainsi, une fois la configuration initiée (soit manuellement, soit lors de la première exécution de l'application), il est possible de l'ajuster en fonction de besoins particuliers, via l'onglet Classpath. Un Class Path est donc partie intégrante d'une configuration d'exécution d'un programme Java, au sein d'Eclipse.

Remarque : Dans Eclipse, il ne faut pas confondre Class Path et Build Path. Le Build Path concerne uniquement l'étape de compilation d'un programme Java. Le principe est similaire à celui du Class Path,

puisque en effet, un Class Path est par défaut élaboré à partir du Build Path. Cf. Qu'est-ce que le Build Path d'un projet Java ?

Comment changer les répertoires source et de compilation ?

Auteurs : L'équipe Eclipse , Laurent.B ,

Pour modifier les répertoires source et de compilation :

- Faites un clic droit sur le projet, menu **Properties > Java Build Path**, puis sélectionnez l'onglet **Source**.

Ensuite, vous avez les choix suivants :

- Sélectionnez **Add folder** pour ajouter un répertoire source supplémentaire.
- Sélectionnez **Remove folder** pour ne plus considérer un répertoire comme un répertoire source.
- Dans la zone **Default output folder**, vous pouvez changer le répertoire de compilation par défaut.

Comment ajouter un JAR à mon projet ?

Auteurs : L'équipe Eclipse ,

Pour ajouter un jar qui se trouve dans le workspace :

- Clic droit sur le projet, menu **Properties > Java Build Path**, onglet **Libraries**, puis bouton **Add jar**

Pour un ajouter un jar hors du workspace :

- Clic droit sur le projet, menu **Properties > Java Build Path**, onglet **Libraries**, puis bouton **Add external jar**

Comment ajouter des bibliothèques à mon projet ?

Auteurs : L'équipe Eclipse , Laurent.B ,

- Clic droit sur le projet, menu **Properties > Java Build Path**, onglet **Libraries**, puis bouton **Add Library**

Une **Library** est à considérer ici comme un ensemble de bibliothèques (jars) prédéfini. Elle peut être composée des jars nécessaires à une **JRE**, à un environnement **JEE**, à d'autres environnements spécifiques ou encore, à un environnement issu d'une **User Library** que vous pouvez composer vous-même.

Comment ajouter des fichiers Java compilés '.class' à mon projet ?

Auteurs : L'équipe Eclipse , Laurent.B ,

Pour ajouter un répertoire de fichiers Java compilés '.class' se trouvant dans le workspace :

- Clic droit sur le projet, menu **Properties > Java Build Path**, onglet **Libraries**, puis bouton **Add Class Folder**

A noter que la plupart du temps, les fichiers .class sont packagés dans des jars et donc ce sont les jars qui les possèdent qui doivent être importés. Les fichiers Java compilés d'un même projet, n'ont a priori aucune raison d'être référencés par ce moyen-là. Cette fonctionnalité servira surtout si on a des fichiers .class qui viennent d'ailleurs et dont les sources auraient été perdus ou qu'ils ne nous auraient pas été fournis.

[3.4] Comment ajouter des .class externes à mon projet ?

Auteurs : [djo.mos](#) , [Laurent.B](#) ,

Pour ajouter un répertoire de .class (fichiers Java compilés) qui se trouve en dehors du workspace :

- **Clic droit sur le projet, menu Properties > Java Build Path, onglet Librairies, puis bouton Add External Class Folder**

Cette solution ne s'applique qu'à partir d'Eclipse Ganymede 3.4.

A noter que la plupart du temps, les fichiers .class sont packagés dans des jars et donc ce sont les jars qui les possèdent qui doivent être importés. Les fichiers Java compilés d'un même projet, n'ont a priori aucune raison d'être référencés par ce moyen-là. Cette fonctionnalité servira surtout si on a des fichiers .class qui viennent d'ailleurs et dont les sources auraient été perdus ou qu'ils ne nous auraient pas été fournis.

Comment relier deux projets entre eux ?

Auteurs : [Laurent.B](#) ,

Lorsque dans Eclipse, on a par exemple deux projets A et B, où A a besoin de B pour compiler et s'exécuter, il peut être intéressant de les mettre en relation de façon explicite.

Pour référencer B au sein de A, il suffit de faire un clic droit sur le projet A, menu Properties > Java Build Path, puis onglet Projects.

Ensuite, il faut cliquer sur Add... et sélectionner le projet B qui doit apparaître dans la liste proposée.

Cette relation doit être à sens unique car B n'est pas censé avoir besoin de A pour fonctionner, ou alors c'est qu'il y a très probablement un problème d'architecture dans votre application.

L'avantage principal qu'un projet A référence un projet B, est tout simplement que cela évite :

- La compilation et le packaging (jar) du projet B pour pouvoir en faire usage dans le projet A. Les modifications de B sont donc instantanément disponibles dans A.
- La Javadoc de l'API du projet B est implicitement disponible, ainsi que bien évidemment son code source (inutile de passer par CTRL + SHIFT + T pour aller à la déclaration d'une classe de B).
- Aucun paramétrage supplémentaire n'est en principe nécessaire pour exécuter A, le classpath est déjà en phase.

[2.x] Comment générer les getters et setters d'une classe ?

Auteurs : [zork](#) ,

Pour générer automatiquement les getters et setters d'une classe, il suffit de faire :

- **Clic droit sur la classe concernée (dans le Package Explorer ou l'éditeur Java), puis sélectionner le menu Source > Generate Getters and Setters**

Le menu est également accessible par le raccourci CTRL + SHIFT + S

Dans l'éditeur, vous pouvez surligner une variable afin que celle-ci soit présélectionnée dans l'assistant de génération des getters/setters.

[3.x] Comment générer les getters et setters d'une classe ?

Auteurs : [djo.mos](#) ,

Pour générer automatiquement les getters et setters d'une classe, il suffit de faire :

- **Clic droit sur la classe concernée (dans le Package Explorer ou l'éditeur Java), puis sélectionner le menu Source > Generate Getters and Setters**

Le menu est également accessible par le raccourci ALT + SHIFT + S

Dans l'éditeur, vous pouvez surligner une variable afin que celle-ci soit présélectionnée dans l'assistant de génération des getters/setters.

Comment placer facilement un point-virgule à la fin de l'instruction ?

Auteurs : [bobuse](#) , [Laurent.B](#) ,

Il faut configurer le positionnement automatique du point-virgule :

Allez dans le menu Window > Preferences > Java > Editor, déployez typing et cochez Semicolons dans Automatically insert at correct position.

Il suffit ensuite de taper un point-virgule et quelle que soit la position du curseur dans l'instruction courante, il sera ajouté à la fin.

Comment aller d'une accolade à l'autre ?

Auteurs : [Lunatic](#) ,

**Pour aller d'une accolade { à l'autre } vous pouvez utiliser le raccourci clavier CTRL + SHIFT + P
De plus un double-clic sur une accolade surligne tout le bloc jusqu'à l'accolade fermante.**

lien : [FAQ](#) Les raccourcis claviers indispensables pour la programmation Java

Comment organiser mes imports ?

Auteurs : [L'équipe Eclipse](#) , [Laurent.B](#) ,

Sur un fichier Java en cours d'édition, faites un clic droit, puis sélectionnez le menu Source > Organize Imports. Ceci correspond au raccourci clavier CTRL + SHIFT + O.

Cette commande est également applicable dans la vue Package Explorer, directement sur un package Java ou encore sur une sélection multiple de fichiers Java (à condition que le(s) fichier(s) et/ou répertoire(s) fassent partie d'un répertoire source).

Si la résolution automatique rencontre des homonymes, par exemple `java.util.List` et `java.awt.List`, une boîte de dialogue vous demandera de sélectionner la bonne référence.

A noter que la réorganisation des imports permet de déclarer explicitement chacune des classes (donc sans l'astérisque `'x.y.*'`), tout en ajoutant celles qui n'auraient pas encore été importées. La réorganisation trie également les imports selon les packages et l'ordre alphabétique des classes importées.

lien : [FAQ](#) Les raccourcis claviers indispensables pour la programmation Java

[3.1-] Comment développer en Java 5.0 ?

Auteurs : [L'équipe Eclipse](#) ,

Pour utiliser les nouveautés de Java 5 (autoboxing, generics etc..) sous Eclipse, vous avez deux solutions :

- Utiliser Eclipse 3.1 ou supérieur
- Si vous avez une version antérieure à Eclipse 3.1, vous pouvez utiliser un plug-in qui gère les nouveautés de Java 5.

Pour le télécharger, cf.  <http://www.genady.net/forum/viewtopic.php?t=31>.

Il faut ensuite l'installer en le dézippant dans le sous-répertoire plugins d'Eclipse et le configurer comme indiqué ici  <http://www.genady.net/forum/viewtopic.php?t=59>

Comment lier la Javadoc au JDK ou à une lib ?

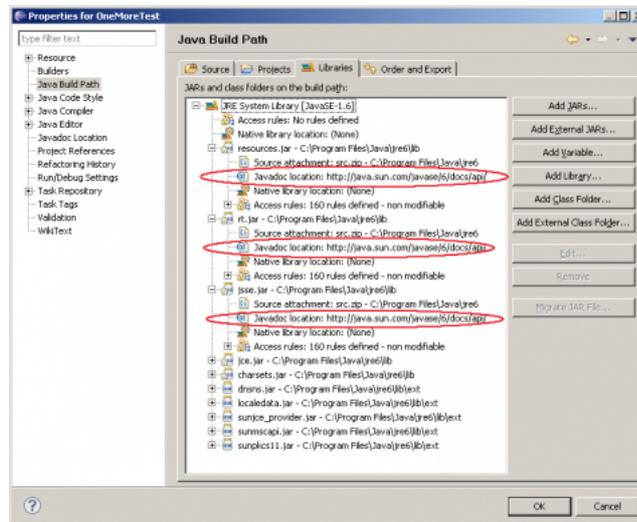
Auteurs : [L'équipe Eclipse](#) , [Laurent.B](#) ,

Un JRE (Java Runtime Environment) n'embarque pas la Javadoc ou les sources de l'API standard de Java, donc si vous en avez besoin (ce qui est quasiment obligatoire lorsque l'on développe en Java), il faut avoir installé un JDK (Java Development Kit). Une fois que c'est fait, vous pourrez constater la présence du fichier `src.zip`, lequel contient tous les fichiers source (`.java`) de l'API standard. Et de fait, dans ces fichiers source, vous aurez également la Javadoc.

Donc, en référençant un JDK, automatiquement le fichier `src.zip` donne accès à la Javadoc. Concrètement, cela signifie qu'en survolant les classes de l'API standard, ou en affichant la view Javadoc, vous devriez pouvoir visualiser la Javadoc, voire même d'ouvrir le code source de cette API (via la touche F3 ou le CTRL + Clic gauche).

Si vous n'avez pas de JDK installé, que vous ne pouvez pas en installer, ou que pour X raison vous devez vous contenter d'un JRE, alors pour obtenir la Javadoc de l'API standard de Java malgré tout, vous avez les possibilités suivantes :

- En principe, dans les versions récentes d'Eclipse, vous ne paramétrez rien du tout, celui-ci sait déjà à quel URL aller chercher la Javadoc de l'API standard.
Pour Java 6, l'URL est <http://java.sun.com/javase/6/docs/api/>, comme le montre ceci.



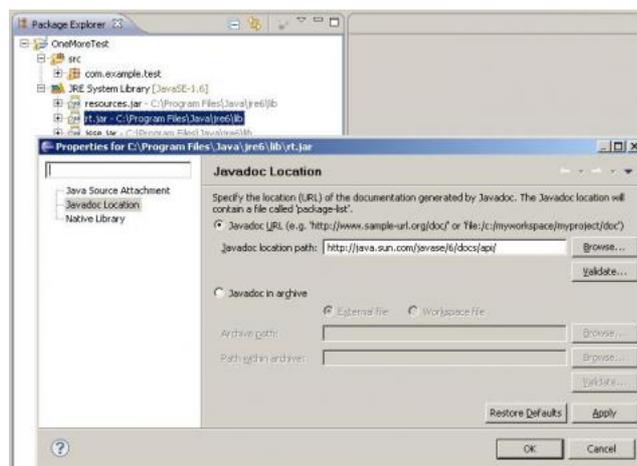
Dans ce cas précis, l'inconvénient est qu'il faut être sûr d'avoir un accès Internet actif, sans quoi, la Javadoc ne sera évidemment pas disponible.

- Vous pouvez aussi vous débrouiller de rapatrier un fichier src.zip, en le plaçant dans le répertoire de votre JRE.

Sur l'image ci-dessus, on peut d'ailleurs voir que cela a été fait ainsi, puisque le Source attachment pointe dessus. De fait, dans ce cas de figure, le paramétrage de la Javadoc location devient inutile mais cela ne pose pas de problème d'avoir les deux a priori.

En revanche, ce qu'il faut évidemment éviter, c'est de paramétrer une version de src.zip qui ne corresponde pas à votre cas de figure.

En principe, un JRE aura toujours la même API d'une version majeur à l'autre, donc l'URL de sa Javadoc ne changera pas, même suite au passage par exemple, d'un JDK/JRE 1.6.0_14 à 1.6.0_20. En revanche, cela sera beaucoup moins vrai pour les sources, donc le fichier src.zip doit vraiment être en phase avec la version de Java utilisée.



L'exemple précédent montre comment configurer la Javadoc ou l'attachement des sources, en passant par les propriétés du projet Java. En réalité, le même paramétrage est également possible via les préférences générales.

- Pour Configurer la Javadoc de l'API standard de Java :
 - Aller dans le menu Window > Preferences > Java > Installed JREs
 - Dans le panneau de droite, sélectionner le JRE

- Cliquer sur le bouton Edit
 - Renseigner l'URL (ou le répertoire, ou l'archive)
- Pour Configurer la Javadoc d'une librairie utilisée dans un projet, par exemple la librairie Struts dont le jar du binaire est struts.jar.
 - Aller dans le menu Project > Properties > Java Build Path > onglet Libraries
 - Sélectionner le fichier struts.jar (ou le jar qui vous intéresse) puis ouvrir l'arborescence de cette librairie
 - Sélectionner l'élément Javadoc location
 - Cliquer sur le bouton Edit
 - Renseigner une URL similaire à ceci struts/docs/api/, qui est la racine de la Javadoc

La configuration de la Javadoc d'un librairie est possible en spécifiant une URL, un répertoire ou une archive.

Il est recommandé de rapatrier toutes les Javadocs distantes dans un répertoire local, afin d'avoir un accès rapide et ne pas être dépendant d'une connexion Internet.

Comment lier les sources au JDK ou à une lib ?

Auteurs : L'équipe Eclipse , Laurent.B ,

- Configurer les sources de l'API standard de Java :
 - Aller dans le menu Window > Preferences > Java > Installed JREs
 - Dans le panneau de droite, sélectionner le JRE
 - Cliquer sur le bouton Edit
 - Renseigner l'URL (ou le répertoire, ou l'archive)

A noter qu'il faut cibler un JDK et non pas un JRE car seul le JDK possède le code source. D'une manière générale, il est toujours préférable de cibler un JDK à cet endroit-là car, en plus de vous permettre de disposer des sources et de la Javadoc, certains outils ne sont utilisables qu'avec un JDK.

- Configurer le code source d'une librairie utilisée dans un projet, par exemple la librairie Struts dont le jar du binaire est struts.jar.
 - Aller dans le menu Project > Properties > Java Build Path > onglet Libraries
 - Sélectionner le fichier struts.jar (ou le jar qui vous intéresse) puis ouvrir l'arborescence de cette librairie
 - Sélectionner l'élément Source attachment
 - Cliquer sur le bouton Edit
 - Renseigner une URL similaire à ceci struts/src/, qui est la racine du répertoire source

La configuration des sources d'une librairie est possible en spécifiant une URL, un répertoire ou une archive.

Il est recommandé de rapatrier tous les packages de sources distants dans un répertoire local, afin d'avoir un accès rapide et ne pas être dépendant d'une connexion Internet.

A noter également qu'en rattachant le code source à une librairie, vous rendez implicitement disponible sa Javadoc.

Lorsque vous voulez consulter les sources d'une classe, par exemple String, appuyer sur SHIFT puis placez votre curseur au-dessus de la déclaration String. Ainsi, vous verrez apparaître une fenêtre présentant le code source de la classe survolée.

Si vous souhaitez réellement "descendre" dans le code source de classe, appuyez à la place sur CTRL puis clic gauche sur la classe ciblée.

[2.x] Comment changer le browser utilisé pour afficher les pages d'aide ?

Auteurs : [bobuse](#) ,

Menu Window > Preferences, puis dans Help, sélectionner l'item Custom browser. Vous pouvez modifier la commande lancée

Par exemple sous KDE, mettre "kfmclient openURL %1". pour ouvrir l'aide dans un nouvel onglet de Konqueror ou "konqueror %1" mais dans ce cas une nouvelle fenêtre est ouverte à chaque fois

[3.x] Comment changer le browser utilisé pour afficher les pages d'aide ?

Auteurs : [djo.mos](#) ,

Menu Window > Preferences, puis dans General, sélectionner l'item "Web Browser".

Vous pouvez soit utiliser le navigateur interne d'Eclipse (en cochant "Use internal web Browser"), soit utiliser un navigateur externe installé dans le système (en cochant "Use external Web browser"). Il faut ensuite choisir l'un des navigateurs installés dans la liste ou en ajouter un via les boutons "Add" ou "Search".

Comment générer la JavaDoc de votre projet ?

Auteurs : [Lunatic](#) , [Laurent.B](#) ,

Bouton droit sur le projet, menu Export... > Java > Javadoc, indiquez la destination et cliquez sur Finish.

- Lors de la première utilisation, vous devrez renseigner l'emplacement de l'utilitaire Javadoc, dans la première boîte de dialogue.
- Vous avez la possibilité de sélectionner spécifiquement les packages/classes/répertoires source, pour lesquels vous souhaitez générer la Javadoc.

Comment exporter son application dans une archive JAR ?

Auteurs : [Lunatic](#) , [djo.mos](#) , [Laurent.B](#) ,

Bouton droit sur le projet, menu Export... > Java > JAR File > Next.

- Dans la 1ère page de l'assistant, vous devez spécifier le nom du fichier jar exporté, mais aussi sélectionner avec précision les classes et ressources à inclure dans le jar.
- Dans la 2ème page, il est possible (entre autres) d'ajouter une description du Jar généré au workspace.
- Dans la 3ème page, il est possible d'indiquer à Eclipse, soit de générer automatiquement le fichier MANIFEST (et éventuellement l'ajouter au workspace), soit d'utiliser un fichier MANIFEST existant (dans le workspace).
Dans cette même page, il est possible de choisir la classe à exécuter automatiquement au lancement du jar.

Attention : Si vous avez ajouté des jars externes au build path du projet, le jar ne pourra pas fonctionner une fois généré car Eclipse n'inclut pas les informations du build path.

Pour pallier cela :

- **Fournir à l'assistant d'export votre propre fichier manifest (généré par un premier export éventuellement) dans lequel vous précisez les jars que vous utilisez en ajoutant la ligne suivante:
Class-Path: log4j.jar, mysql-connector-java-5.1.6-bin.jar**

Assurez-vous également que ces jars soient présents à côté du jar généré, avant de le lancer.

[3.4] Comment exporter son application dans une archive JAR exécutable ?

Auteurs : [djo.mos](#) , [Laurent.B](#) ,

Eclipse 3.4 (Ganymede) offre un nouvel assistant d'export de JAR exécutable, qui inclut toutes les dépendances du projet (JARs externes) dans l'archive générée.

Avant de pouvoir utiliser cet assistant, il faut avoir exécuté le projet au moins une fois, c'est à dire avoir au moins une configuration de lancement.

Ensuite, faites un clic droit sur le projet, menu Export... > Java > Runnable JAR File > Next.

Dans la fenêtre de l'assistant, vous devez sélectionner la configuration de lancement et spécifier le nom du fichier jar exporté, puis valider (bouton "Finish").

Une fenêtre s'affiche vous demandant de revoir les licences des bibliothèques qui seront décompressées et recompressées dans le JAR exporté, pour voir si elles permettent ce cas d'utilisation.

[2.x] Comment passer des arguments à la JVM de mon application ?

Auteurs : [L'équipe Eclipse](#) ,

Bouton droit sur la classe à exécuter dans le package explorer Run as > Run onglet (x)= Arguments

Dans Program arguments : les arguments de l'application (de la méthode main(String[] args))

Dans VM arguments : les paramètres de la machine virtuelle

[3.x] Comment passer des arguments à la JVM de mon application ?

Auteurs : [djo.mos](#) ,

Après avoir lancé l'application au moins une fois pour avoir une configuration de lancement, allez dans le menu Run > Run Configurations et sélectionnez la configuration qui vous intéresse parmi la liste et activez l'onglet onglet (x)= Arguments

Dans Program arguments : les arguments de l'application (de la méthode main(String[] args))

Dans VM arguments : les paramètres de la machine virtuelle

Comment empêcher que mes fichiers du répertoire de compilation soient effacés ?

Auteurs : Lunatix , djo.mos , Laurent.B ,

Parfois, on peut être tenté de placer des fichiers de configuration, dans le répertoire de compilation (application.properties, fichiers .xml, ...).

Cependant, par défaut Eclipse vide ce répertoire lors d'une compilation complète (lorsque l'on sélectionne par exemple le menu Project > Clean...).

Il est possible de paramétrer Eclipse pour qu'il ne fasse plus ce ménage, via le menu Window > Preferences > Java > Compiler > Building > Output folder, en décochant Scrub output folder when cleaning projects.

Cependant, une solution bien plus propre et largement préférable est de créer un répertoire dans lequel vous ne placez que vos fichiers de configuration.

Donc, tout simplement :

- Créez un répertoire nommé par exemple config au sein du projet.
- Faites ensuite un clic droit dessus, puis sélectionnez le menu Build Path > Use as Source Folder.

Ainsi, le contenu de votre répertoire sera automatiquement pris en compte et recopié dans le répertoire de compilation, à côté de vos fichiers compilés.

A noter que les fichiers du répertoire sont copiés dans le répertoire de compilation par défaut, qui est généralement le répertoire /bin. Vous pouvez cependant changer de répertoire de destination, si cela vous est nécessaire, en le paramétrant dans les propriétés du projet (cf. Comment changer les répertoires source et de compilation ?).

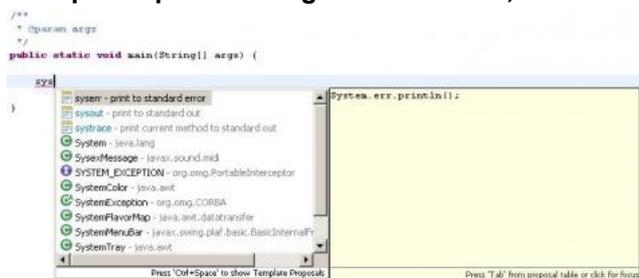
Comment utiliser ou définir des templates de code ?

Auteurs : bobuse , Laurent.B ,

Les templates sont des bouts de code accessibles via des raccourcis typographiques.

Par exemple, tapez sys quelque part dans votre code source d'une classe Java et appuyez sur CTRL + Espace.

Ceci provoque l'affichage d'une fenêtre, comme indiqué sur l'image suivante :



Sélectionnez sysout - print to standard out et appuyez sur Entrée.

Et voilà ! Le code System.out.println(); a été inséré.

Parmi les templates prédéfinis, ceux-ci vous permettent de générer des squelettes de code qui vous aideront à programmer plus rapidement.

Autres exemples :

- **Créer une instance d'une classe :**
 - Tapez new puis CTRL + Espace
 - Sélectionnez le template new - create new Object et le code type name = new type(arguments); est généré, avec les champs à compléter encadrés, et le premier mot type sélectionné.
 - Sans déplacer le curseur, saisissez Integer en lieu et place du mot type et vous constaterez que le troisième champ (l'appel du constructeur) se modifie simultanément !
 - Appuyez sur Tab pour passer au champ suivant (le nom de la variable) et donnez le nom que vous souhaitez à votre variable (par exemple : monInteger).
 - Appuyer de nouveau sur Tab et vous passez à l'argument du constructeur de Integer, dans lequel vous pouvez initialiser votre variable (par exemple : 10).
 - Pour finir, appuyer sur Entrée et votre votre variable est initialisée et les cadres entourant les éléments à adapter disparaissent.
- **Générer rapidement une boucle d'itération :**
 - Tapez for puis CTRL + Espace et complétez les éléments encadrés

Vous pouvez également définir vos propres templates :

Pour cela, allez dans le menu Window > Preferences > Java > Editor > Templates.

Pour des besoins spécifiques, analysez ceux qui sont déjà présents pour comprendre comment produire les vôtres.

L'outillage fourni est relativement puissant mais il faut apprendre à le maîtriser avant de savoir si vous pourrez créer les templates souhaités.

Voici un exemple de création de template manquant, qui permet de formater plus proprement un texte de Javadoc, en ajoutant les balises HTML <p></p> autour d'un paragraphe :

- Cliquez sur le bouton New... et remplissez les valeurs qui suivent
- Name : <p>
- Context : Javadoc
- Description : <p></p>
- Pattern :

```
<p>${word_selection}${}</p>${cursor}
```

A noter que ce template est basé sur celui du qui met en gras une zone de texte Javadoc. Malheureusement, (sauf erreur et évolutions ultérieures) les limitations de l'outil imposent que l'on ne puisse appliquer ce template que sur un texte de Javadoc d'une seule ligne. Donc, cela nécessite de manipuler manuellement le paragraphe pour que les balises début et fin, soient placées aux bons endroits...

Un autre exemple très utile lorsque vous utilisez l'api commons-logging, de manière à générer proprement et dynamiquement la déclaration du logger :

- Cliquez sur le bouton New... et remplissez les valeurs qui suivent
- Name : lg
- Context : Java
- Description : Instanciation de l'objet de log
- Pattern :

```
/** Traces */  
private static final Log log = LoggerFactory.getLog(${enclosing_type}.class);
```

A utiliser en début de corps d'une classe. Très efficace contre les mauvais copier/coller qui gardent malencontreusement, et à fort mauvais escient, le nom de la classe d'origine...

Remarque : Les templates sont importables et exportables, ce qui est notamment pratique pour homogénéiser le développement d'applications en équipe.

Ils ne sont pas réservés au langage Java (et autres langages liés) mais pour pouvoir les utiliser dans d'autres langages/technos, il faut qu'un menu dédié soit disponible dans les préférences.

C'est l'ajout de plug-ins embarquant ce genre de fonctionnalité qui peut permettre d'en disposer.

Comment utiliser JUnit avec Eclipse ?

Auteurs : L'équipe Eclipse ,

Le framework de test Junit est très utilisé et est disponible sur  www.junit.org
Il est complètement intégré à Eclipse, et vous n'avez besoin de rien installer.

Dans le package explorer bouton droit sur une classe a tester > New Junit test case et vous aurez un assistant déjà renseigné.

- La deuxième page de l'assistance permet de sélectionner les méthodes et constructeurs à tester
- Je vous conseille de placer les classes Junit dans le même package que la classe à tester (fait par défaut), mais dans un répertoire de sources différent
- Si vous codez d'abord la classe de test (façon xp programming), faites file > new > junit test case

Pour exécuter le ou les tests, bouton droit sur la classe ou le package, Run as > JUnit test

Comment debugger une application Java ?

Auteurs : Lunatix , Laurent.B ,

Dans la marge gauche d'une classe, vous pouvez ajouter (ou retirer) un point d'arrêt, en double cliquant sur la ligne souhaitée. A noter que vous obtiendrez le même résultat en passant par un clic droit > Toggle breakpoint.

Ce point d'arrêt entre en action lorsque vous exécutez votre programme en mode Debug et que la ligne où il se trouve doit être exécutée.

Eclipse gèle alors l'exécution du programme et attend une action de l'utilisateur pour savoir comment le débogage doit continuer.

Vous pouvez mettre des breakpoints partout ou cela vous semble nécessaire mais uniquement à des endroits cohérents.

Les actions possibles du mode Débug :

- Dans la view Debug : Step into (F5) permet de rentrer dans une méthode
- Dans la view Debug : Step over (F6) permet de poursuivre l'exécution en mode pas à pas.
- Dans la view Debug : Step return (F7) permet de poursuivre l'exécution jusqu'à sortir de la méthode courante.
- Dans la view Debug : Resume (F8) permet de reprendre l'exécution normalement ou s'arrêter uniquement si un autre point d'arrêt est rencontré.
- Dans la view Variables : Vous pouvez consulter les valeurs des variables
- Dans la view Variables : Une variable en rouge est une variable qui vient d'être modifiée

- **Dans la view Variables : Bouton droit sur une variable : Edit value permet de changer à chaud la valeur**
- **Dans la view Breakpoints : Vous pouvez activer/désactiver des breakpoints sans les retirer**
- **Dans la view Breakpoints : Le bouton j! permet de mettre un breakpoint sur toutes les Exceptions du programme**
- **Dans la view Breakpoints : à l'aide de la flèche noire orientée vers le bas, activez Show working Sets. Puis dans le même menu, sélectionnez Working Sets, Exception et donnez un nom. Voilà, vous avez maintenant un groupe d'Exceptions sous ce nom, que vous pouvez activer d'un simple clic de souris.**

Remarque : Si jamais vous constatez qu'un point d'arrêt, lorsque vous tentez de le placer sur une ligne de code, se retrouve décalé, alors cela signifie très probablement que le répertoire de compilation configuré dans votre projet, n'est pas à jour par rapport au code source affiché. Cela est souvent le cas lorsque par exemple vous compilez avec Ant et que les répertoires des classes compilées ne sont pas identiques à ceux paramétrés dans votre projet Eclipse.

De même, si en cours de debug, vous constatez que le mode pas à pas n'est pas synchronisé avec le code source affiché, c'est sans doute pour les mêmes raisons.

Si vous ne compilez pas avec Ant ou un autre outil de build, vérifiez que l'option Build Automatically est cochée dans le menu Project.

Sommaire > Développement J2EE

[2.x] Comment développer en JEE avec Eclipse ?

Auteurs : [L'équipe Eclipse](#) ,

Cette version d'Eclipse n'intègre pas par défaut les outils nécessaires au développement J2EE. Il faut donc installer un ou plusieurs plug-ins supplémentaires.

Par exemple :

- **Plug-in Tomcat**  www.eclipsetotale.com/tomcatPlugin.html
- **Eclipse Web tools**  www.eclipse.org/webtools/
- et d'autres...

[3.x] Comment développer en JEE avec Eclipse ?

Auteurs : [djo.mos](#) ,

Il est possible de développer en JEE avec Eclipse, en partant par exemple du package "Eclipse for JavaEE Developers" ou alors, en installant  **WebTools**.

Ainsi, on a la possibilité d'utiliser les outils suivants :

- Assistants pour la génération d'applications Web (WAR) ou EAR.
- Assistants pour la création de Servlets, Filters, EJBs, etc.
- Editeur de fichiers HTML, JSP, CSS et JS.
- Support natif des serveurs les plus populaires (Tomcat, JBoss, WebSphere, JOnAs, OC4J, etc.)
- Support de JSF
- Support de JPA
- etc.

[2.x] Que sont les Web Tools ?

Auteurs : [Lunatix](#) , [Laurent.B](#) ,

Il s'agit des outils standards de développement JEE pour Eclipse.

Le site des Web Tools est  [Projet Eclipse Web Tools \(WTP\)](#)

[2.x] Comment installer les Web Tools ?

Auteurs : [L'équipe Eclipse](#) ,

Le plug-in Web Tools (WTP) est le plug-in officiel pour le développement JEE. Vous pouvez l'installer très simplement via l'Update Manager.

Menu Help > Software updates > find and Install > Search for new features to install > deployez Eclipse.org update site > sélectionnez WTP , GEF, JEM et EMF.

Il ne vous reste plus qu'à accepter les licences et valider l'installation.

Attention, tous les miroirs ne disposent pas des WTP.

Plus de détails sur la procédure sur le site d'Eclipse  [update Site](#).

[3.3, 3.2] Comment installer Eclipse WebTools ?

Auteurs : [djo.mos](#) ,

Le plug-in Web Tools Project (WTP) est le plug-in officiel pour le développement JEE. Vous pouvez l'installer très simplement à l'aide du gestionnaire de mises à jour (si besoin consultez la [FAQ](#) [3.3-] Comment installer un plug-in ?).

Assurez-vous d'abord que l'update-site de WTP est bien dans la liste des sites inclus dans la recherche. S'il n'est pas présent, cliquez sur le bouton Add Remote Site, puis donnez un nom et renseignez l'URL <http://download.eclipse.org/webtools/updates/>

L'installation se fait à l'aide du menu Help > Software updates > Find and Install > Search for new features to install, déployez Eclipse.org update site, sélectionnez WTP et ses dépendances. Il ne vous reste plus qu'à accepter les licences et valider l'installation.

Une autre solution consiste à télécharger le package Eclipse IDE for Java EE Developers, qui intègre directement WTP et qui est disponible sur la page  <http://www.eclipse.org/downloads/moreinfo/jee.php>.

[3.4] Comment installer Eclipse WebTools ?

Auteurs : [djo.mos](#) ,

Le plug-in Web Tools Project (WTP) est le plug-in officiel pour le développement JEE. Vous pouvez l'installer très simplement à l'aide du gestionnaire de mises à jour (si besoin consultez la [FAQ](#) [3.4] Comment installer un plug-in ?).

Assurez-vous d'abord que l'update-site de WTP est bien dans la liste des sites inclus dans la recherche. S'il n'est pas présent, cliquez sur le bouton Add Remote Site, puis renseignez l'URL <http://download.eclipse.org/webtools/updates/>

L'installation se fait à l'aide du menu Help > Software updates > Available Software > Search for new features to install, déployez Webtools (WTP) Update Site, sélectionnez WTP et ses dépendances. Il ne vous reste plus qu'à accepter les licences et valider l'installation.

Une autre solution consiste à télécharger le package Eclipse IDE for Java EE Developers, qui intègre directement WTP et qui est disponible sur la page  <http://www.eclipse.org/downloads/packages/release/ganymede/sr2>.

[3.5] Comment installer Eclipse WebTools ?

Auteurs : [djo.mos](#) ,

Le plug-in Web Tools Project (WTP) est le plug-in officiel pour le développement JEE. Vous pouvez l'installer très simplement à l'aide du gestionnaire de mises à jour (si besoin consultez la [FAQ](#) [3.5] Comment installer un plug-in ?).

Assurez-vous d'abord que l'update-site de WTP est bien dans la liste des sites inclus dans la recherche. S'il n'est pas présent, cliquez sur le bouton Add Remote Site, puis renseignez l'URL <http://download.eclipse.org/webtools/updates/>

L'installation se fait à l'aide du menu Help > Install New Software..., sélectionnez The Eclipse Web Tools Platform (WTP) Project update site - <http://download.eclipse.org/webtools/updates/>, puis cochez les composants WTP qui vous intéressent, en particulier le bloc Web Tools Platform (WTP) 3.1.0. Il ne vous reste plus qu'à accepter les licences et valider l'installation.

Une autre solution consiste à télécharger le package Eclipse IDE for Java EE Developers, qui intègre directement WTP et qui est disponible sur la page <http://www.eclipse.org/downloads/>.

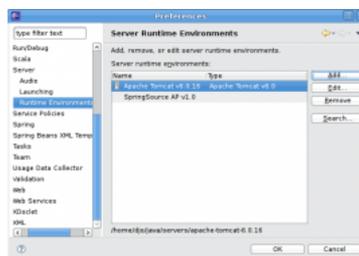
[3.2+] Comment ajouter un serveur à Eclipse ?

Auteurs : [djo.mos](#) ,

Eclipse WebTools offre un support pour une multitude de serveurs d'applications Java, qui permet entre autres de deployer directement une application sur le serveur depuis Eclipse.

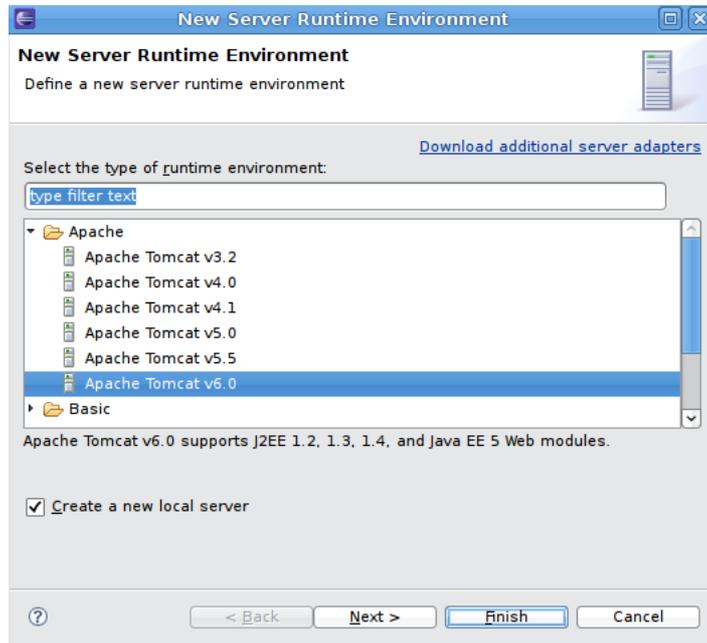
Pour ajouter un nouveau serveur à Eclipse, Allez dans :

- Eclipse 3.3 ou antérieur : Menu Window > Preferences > Server > Runtime Environments
- Eclipse 3.4 : Menu Window > Preferences > Server > Installed Runtimes



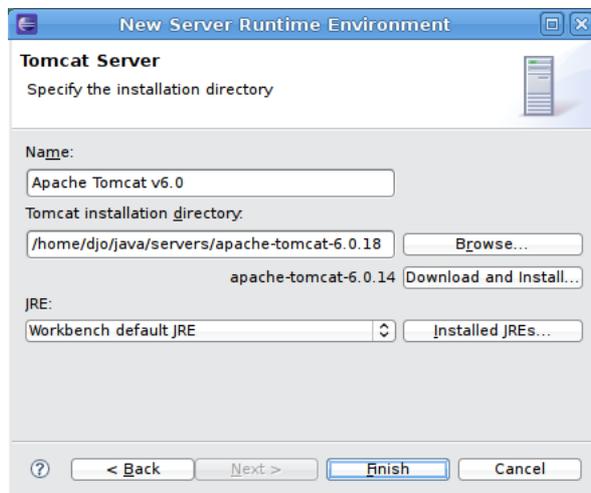
La page de configuration des serveurs

Cliquez sur le bouton "Add". Dans la fenêtre qui s'affiche, sélectionnez le type de serveur et cochez la case "Create a new local server". Cliquez ensuite sur "Next".



La page de configuration du type d'un nouveau serveur

Remplissez les différents champs de la fenêtre (principalement le nom du serveur et son emplacement sur le disque) et validez.



La page de configuration d'un nouveau serveur

Le serveur est désormais configuré et prêt à être utilisé.

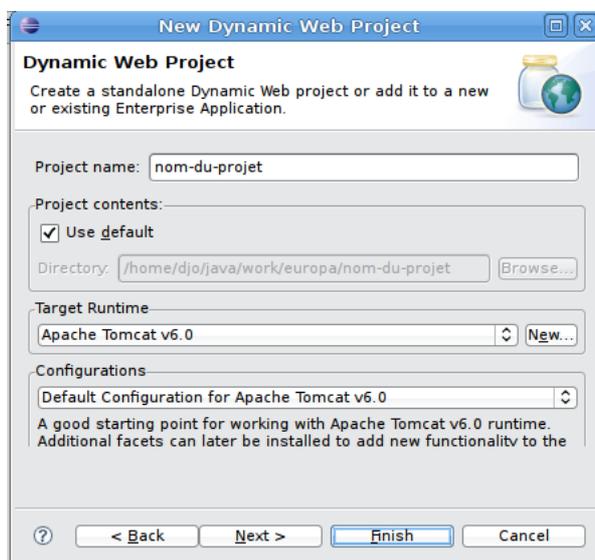
Veillez noter que la configuration relative à ce serveur est stocké dans un nouveau projet crée automatiquement par Eclipse et nommé "Servers". Pour pouvoir utiliser le serveur, il est impératif que ce projet soit ouvert.

[3.2+] Comment créer un projet web Java ?

Auteurs : [djo.mos](#) ,

Pour créer un nouveau projet web Java dans Eclipse, Allez dans Menu File > New > Other > Web > Dynamic Web Project

Dans la fenêtre qui s'affiche, saisissez un nom pour le projet à créer et cliquez sur "Next"



Assistant de création d'un projet web Java

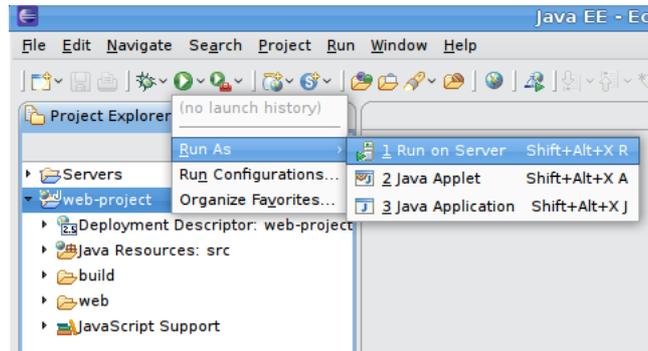
Les pages suivantes permettent de configurer en détail les paramètres du projet. Vous pouvez les laisser tel quels et utiliser les réglages par défaut.

[3.2+] Comment deployer et exécuter une application web sur un serveur ?

Auteurs : [djo.mos](#) ,

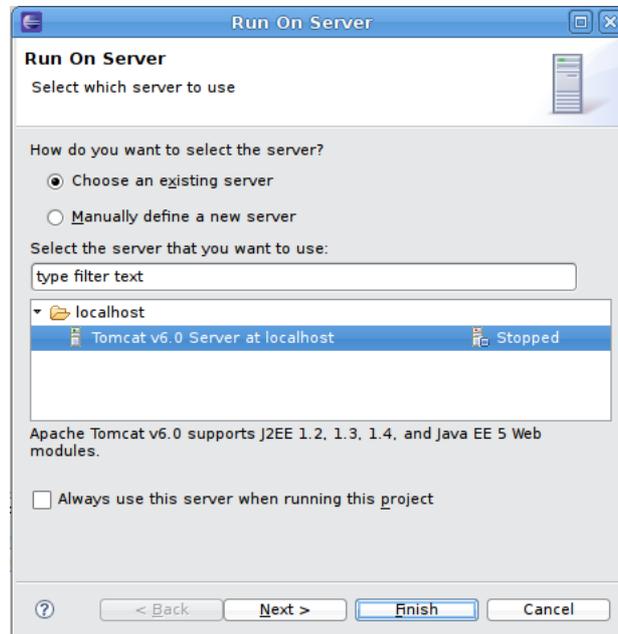
Pour deployer de exécuter un projet web Java dans un serveur depuis Eclipse, Sélectionnez le projet dans la vue "Package Explorer" ou "Project Explorer", puis cliquez sur la petite flèche orienté vers le bas du bouton Run (Cercle vert avec un triangle blanc à l'intérieur" dans le toolbar.

Sélectionnez Run As > Run on Server



Assistant d'exécution d'un projet dans un serveur

Dans la fenêtre qui s'affiche, sélectionnez un serveur déjà configuré et cliquez sur "Finish"



Assistant d'exécution d'un projet dans un serveur

Veillez noter qu'il est possible de créer un nouveau serveur directement depuis l'assistant d'exécution. Pour ce faire, il faut sélectionner l'option "Manually define a new Server" et suivre l'assistant.

Une fois l'assistant terminé, l'application sera déployée sur le serveur et ce dernier démarré.

Sommaire > Développement J2ME

Comment développer en J2ME sous Eclipse ?

Auteurs : Lunatix ,

Utilisez le plug-in EclipseMe :  eclipseme.org
Suivez bien les instructions d'installation sur le site.

[Sommaire](#) > La Plate-Forme Eclipse

Qu'est-ce que la Plate-Forme Eclipse ?

Auteurs : [Laurent.B](#) ,*Traduction de la définition suivante*  <http://www.eclipse.org/platform/overview.php>

La Plate-Forme Eclipse fournit les frameworks et services de base à partir desquels tous les plug-ins d'extensions sont créés. Elle fournit également l'environnement d'exécution dans lequel les plug-ins sont chargés, intégrés et exécutés.

L'objectif principal de la plate-forme est de permettre aux développeurs d'outils tiers de facilement développer et proposer des outils intégrés.

La plate-forme en elle-même est une sorte de plate-forme d'outils universelle. C'est un EDI de tout et de rien en particulier. Elle peut manipuler n'importe quel type de ressource (fichiers Java, C, Word, HTML, JSP, etc) de manière générique, sans pour autant savoir comment les traiter spécifiquement. La plate-forme Eclipse en elle-même n'est pas pourvue d'énormément de fonctionnalités utilisateur, c'est quelque part ce qui fait sont intérêt. Sa véritable valeur vient des plug-ins qu'on lui ajoute afin de lui apprendre à travailler avec différents types de ressources. Cette architecture par plug-in permet une expérience plus transparente pour l'utilisateur final, lorsqu'il navigue entre les différents outils, comme cela n'a jamais été possible auparavant.

La plate-forme Eclipse définit un ensemble de frameworks et de services communs qui forment collectivement une application d'intégration, nécessaire au support d'une plate-forme complète d'intégration. Ces services et frameworks représentent les fonctionnalités communes requises par la plupart des outils de construction, incluant un environnement utilisateur standard, un modèle de projet pour la gestion de ressources, des widgets nativement portables, des bibliothèques d'interface utilisateur, une gestion des modifications de ressources de manière automatique pour les compilateurs et constructeurs, une infrastructure de débogage indépendante du langage, ainsi qu'une infrastructure pour la gestion des ressources versionnées en environnement distribué multi-utilisateur.

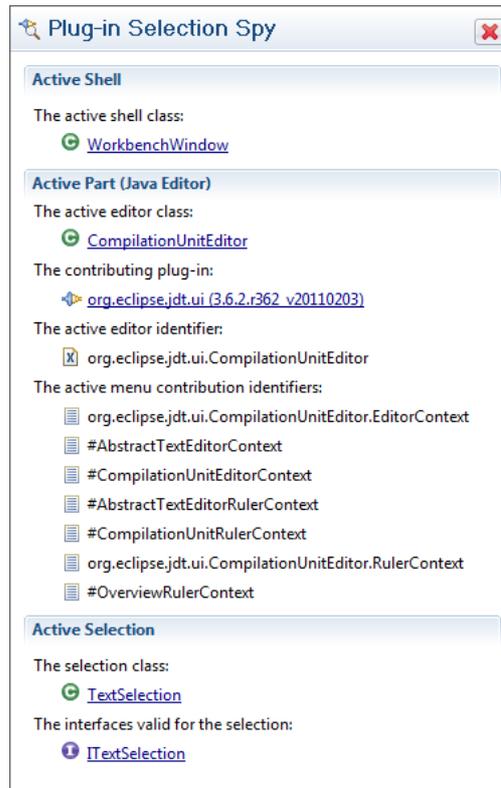
De plus, la plate-forme Eclipse définit un environnement utilisateur et un ensemble de paradigmes d'interactions utilisateur spécifiques à chaque domaine, auquel les outils de construction se connectent pour ajouter de nouvelles capacités. La plate-forme vient avec un ensemble de vues standards qui peuvent être étendues par les outils de construction. Ces derniers peuvent à la fois ajouter de nouvelles vues ainsi que connecter, aux vues existantes, de nouvelles capacités spécifiques à un domaine.

[3.5+]Comment trouver une classe correspondant à un élément Eclipse (menu, action, wizard...)?Auteurs : [Gueritarish](#) ,

Pour pouvoir examiner une classe de la plateforme Eclipse correspondant à un élément affiché directement à l'écran, il faut passer par le plug-in Spy. Cette fonction est accessible à partir de 2 raccourcis clavier :

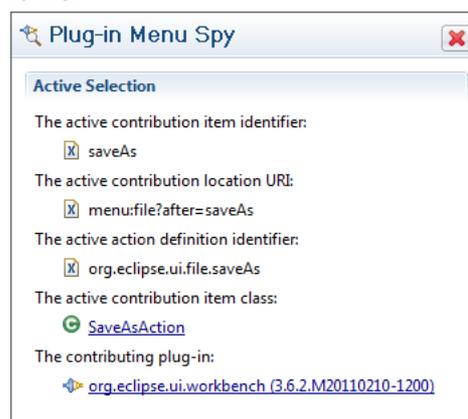
- **ALT + SHIFt + F1 (Plug-in Selection Spy)** permet d'afficher les informations sur la sélection courante (typiquement, le composant ayant le focus: wizard, éditeur, vue..). Ces informations donnent le plug-in contribuant à ce composant, l'identifiant de la (des) section(s) d'aide, les classes implémentant ce composant...

Exemple pour l'éditeur Java:



- **ALT + SHIFT + F2 (Plug-in Menu Spy) permet d'afficher les informations sur la sélection de l'utilisateur. On peut donc chercher des informations sur les commandes dans les menus (contextuels ou non), les toolbars... Les informations contiennent en plus l'emplacement de la commande (entre autre).**

Exemple pour la commande Save As:



Il est à noter que le Plug-in Menu Spy n'est apparu qu'à partir de la version 3.5 d'Eclipse.

[Sommaire](#) > [La Plate-Forme Eclipse](#) > [Développement RCP](#)

Comment développer une application RCP ?

Auteurs : [Laurent.B](#) ,

Pour développer une application RCP, la première chose à faire est de partir du bundle Eclipse for RCP and RAP Developers, disponible ici  <http://www.eclipse.org/downloads/>.

Ensuite, vous pouvez suivre les tutoriels présents ici  <http://eclipse.developpez.com/cours/?page=platform-cat#plugin-dev> ou ici  <http://www.eclipse.org/resources/>.

Lorsque l'on développe une application RCP, un des principaux objectifs est de réutiliser ce qui est déjà disponible, afin bien entendu de ne pas réinventer la roue. Tout ceci nécessite de prendre le temps d'étudier un minimum l'architecture de la plate-forme et ensuite d'identifier les éléments de base nécessaires au développement de l'application en particulier.

Une application RCP étant constituée de plug-ins, il est donc important de savoir comment les architecturer, les implémenter, ainsi que de comprendre leur cycle de vie et leurs dépendances avec le reste de l'application.

Afin de faciliter votre prise en main de la plate-forme Eclipse, n'hésitez pas à vous procurer des livres. Voici quelques-uns des ouvrages disponibles :

-  [Eclipse : Principes, patterns et plug-ins\(21 juillet 2004\)](#)
-  [Eclipse Rich Client Platform \(2nd Edition\)\(22 mai 2010\)](#)
-  [Eclipse Plug-ins \(3rd Edition\)\(21 décembre 2008\)](#)
-  [EMF: Eclipse Modeling Framework \(2nd Edition\)\(26 décembre 2008\)](#)
-  [OSGi and Equinox: Creating Highly Modular Java Systems\(24 février 2010\)](#)

lien : [FAQ Définition de SWT et de JFace](#)

Comment créer un projet de façon programmatique ?

Auteurs : [Gueritarish](#) ,

Pour créer un projet programmatiquement, il faut tout d'abord ajouter une dépendance vers le plug-in org.eclipse.core.resources et ensuite instancier un IProject:

```
String name = "MonProjet";
IWorkspace workspace = ResourcesPlugin.getWorkspace();
IWorkspaceRoot root = workspace.getRoot();
IProject project = root.getProject(name);
project.create(new NullProgressMonitor());
project.open(new NullProgressMonitor());
```

Il faut ensuite ajouter la nature du projet (Java, C++, Android, ...). Généralement, des constantes sont prévues à cet effet.

```
IProjectDescription desc = project.getDescription();
desc.setNatureIds(new String[] { "MyNatureID" });
project.setDescription(desc);
```

**On peut ensuite définir le classpath et le dossier de sortie de compilation...
Par exemple, pour un projet Java (avec la dépendance vers org.eclipse.jdt.core):**

```
// Dossier de sortie vers "bin":
IJavaProject javaProj = JavaCore.create(project);
IFolder binDir = project.getFolder("bin");
IPath binPath = binDir.getFullPath();
javaProj.setOutputLocation(binPath, null);
// Mise en place du classpath:
String path = "c:\\jre\\lib\\rt.jar";
IClasspathEntry cpe= JavaCore.newLibraryEntry(path, null, null);
javaProj.setRawClasspath(new IClasspathEntry[] {cpe});
```

[Sommaire](#) > [La Plate-Forme Eclipse](#) > [Développement de Plug-ins](#)[Comment développer un plug-in Eclipse ?](#)Auteurs : [Laurent.B](#) ,

Pour développer un plug-in Eclipse, la première chose à faire est de partir du bundle Eclipse for RCP and RAP Developers, disponible ici  <http://www.eclipse.org/downloads/>.

Ensuite, vous pouvez suivre les tutoriels présents ici  <http://eclipse.developpez.com/cours/?page=platform-cat#plugin-dev> ou ici  <http://www.eclipse.org/resources/>.

Afin de faciliter votre prise de connaissances sur le développement de plug-ins, n'hésitez pas à vous procurer des livres.

Voici quelques-uns des ouvrages disponibles :

-  [Eclipse : Principes, patterns et plug-ins](#)(21 juillet 2004)
-  [Eclipse Plug-ins \(3rd Edition\)](#)(21 décembre 2008)

lien : [FAQ Définition de SWT et de JFace](#)

[Comment activer, désactiver, rendre visible ou invisible un menu en fonction d'une variable ?](#)Auteurs : [Gueritarish](#) ,

Tout d'abord, pour créer un menu que l'on peut activer ou désactiver en fonction du contexte, voici comment il faut procéder :

Déclarer un SourceProvider dans le fichier plugin.xml de l'application :

```
<extension point="org.eclipse.ui.services">
  <sourceProvider provider="com.foo.myapp.MySourceProvider">
    <variable
      name="com.foo.myapp.myVariable"
      priorityLevel="workbench">
    </variable>
  </sourceProvider>
</extension>
```

Où :

- `com.foo.myapp.MySourceProvider` est la classe qui va étendre `ISourceProvider`
- `com.foo.myapp.myVariable` est le nom de la variable

Ensuite, créer la classe `MySourceProvider` dans le package `com.foo.myapp`. Cette classe hérite de `AbstractSourceProvider` (pour plus de lisibilité).

```
public class MySourceProvider extends AbstractSourceProvider {

  /** Nom de la variable à vérifier. */
  public final static String VARIABLE = "com.foo.myapp.myVariable";

  /** Valeur de l'état actif. */
  private final static String ENABLED = "enabled";
```

```

/** Valeur de l'état inactif. */
private final static String DISABLED = "disabled";

/** L'état courant de la variable. */
private boolean enabled;

@Override
public String[] getProvidedSourceNames() {

    return new String[] { VARIABLE };
}

@Override
public Map getCurrentState() {

    String variabletState = enabled ? ENABLED : DISABLED;

    Map currentState = new HashMap(1);
    currentState.put(VARIABLE, variabletState);

    return currentState;
}

/**
 * Met à jour l'état de la variable.
 *
 * @param flag Flag fournissant l'état de la variable :
 * <code>>true</code> pour {@link MySourceProvider#ENABLED ENABLED},
 * <code>>false</code> pour {@link MySourceProvider#DISABLED DISABLED}.
 */
public void setVariableState(boolean flag) {

    if(this.enabled == flag) {
        return; // Pas de modification
    }

    this.enabled = flag;
    String currentState = enabled ? ENABLED : DISABLED;
    fireSourceChanged(ISources.WORKBENCH, VARIABLE, currentState);
}
}

```

Dans le fichier plugin.xml, à la définition du handler de la commande, rajouter :

```

<extension
    point="org.eclipse.ui.handlers">
    <handler
        class="com.foo.MyHandler"
        commandId="com.foo.myCommand">
        <activeWhen>
            <with
                variable="com.foo.myapp.myVariable">
                <equals
                    value="enabled">
                </equals>
            </with>
        </activeWhen>
    </handler>
</extension>

```

Ensuite, appeler la méthode `setVariableState()` pour mettre à jour la valeur de la variable (enabled ou non).
Ex. dans un listener :

```

boolean newVariableState = true;
// Récupération du IServiceProviderService.

```

```
ISourceProviderService sourceProviderService = (ISourceProviderService)
PlatformUI.getWorkbench().getService(ISourceProviderService.class);

if (sourceProviderService != null) {

    // Récupération du fournisseur de la variable en question.
    ISourceProvider provider = sourceProviderService.getSourceProvider(MySourceProvider.VARIABLE);

    if (provider instanceof MySourceProvider) {
        // Positionnement du nouvel état de la variable.
        ((MySourceProvider) provider).setVariableState(newVariableState);
    }
}
```

Pour ce qui est de rendre visible ou invisible le menu, on peut utiliser le même principe, c'est-à-dire en déclarant ceci :

```
<extension
    point="org.eclipse.ui.menus">
    <menuContribution
        locationURI="menu:com.foo.my.menu?after=additions">
        <command
            commandId="com.foo.myCommand">
            <visibleWhen
                checkEnabled="false">
                <with
                    variable="com.foo.myapp.myVariable">
                    <equals
                        value="enabled">
                    </equals>
                </with>
            </visibleWhen>
        </command>
    </menuContribution>
</extension>
```

Comment ajouter un JAR à un plug-in Eclipse

Auteurs : Gueritarish , Laurent.B ,

Si l'un de vos plug-ins nécessite l'utilisation d'un JAR spécifique, l'explication suivante va vous indiquer comment intégrer celui-ci. Toutefois, avant de suivre cette procédure, vous devez déterminer le besoin précis, c'est-à-dire si le JAR en question est nécessaire à un seul plug-in ou au contraire à plusieurs.

Si votre composant n'est constitué que d'un seul plug-in ou si vous savez que le JAR ne sera utile qu'à un seul des plug-ins de votre composant, suivez le **cas n°1**.

Si vous souhaitez utiliser le JAR dans plusieurs des plug-ins de votre composant, suivez le **cas n°2**.

cas n°1 :

- Placez le fichier JAR concerné à la racine du projet (ce n'est pas obligatoire mais c'est recommandé et cela simplifie la procédure).
- Ensuite, double-cliquez sur le fichier META-INF/MANIFEST.MF, de votre projet de plug-in. Ceci provoque l'ouverture de l'éditeur des propriétés du plug-in.
- Placez vous dans l'onglet Runtime
- Dans la zone ClassPath, cliquez sur le bouton Add..., sélectionnez le fichier JAR que vous aurez mis à la racine, puis validez.

- Placez-vous ensuite sur l'onglet **Build** et assurez-vous que le fichier **JAR** est bien coché dans la zone **Binary Build**. Ainsi, il pourra être intégré au sein du plug-in, au moment du **Build**.

Vous pouvez alors contrôler que dans l'onglet **MANIFEST.MF** celui-ci comporte effectivement une référence similaire à celle-ci :

```
Bundle-ClassPath: une_bibliotheque.jar
```

Il en va de même pour le fichier **build.properties**, lequel doit quant à lui contenir une référence similaire à ceci :

```
bin.includes = META-INF/,\  
              une_bibliotheque.jar
```

Le **JAR** référencé est à présent prêt à être utilisé au sein du plug-in qui le contient.

cas n°2 :

A partir du moment où vous avez besoin qu'un **JAR** puisse être utilisé par plusieurs de vos plug-ins, le plus judicieux est de créer un plug-in totalement dédié à celui-ci.

Vous avez alors 2 possibilités, soit de réaliser manuellement toute l'opération nécessaire, soit de laisser faire Eclipse par l'intermédiaire d'un assistant.

L'opération n'étant pas particulièrement compliquée, voici tout d'abord comment la réaliser manuellement :

- Toute d'abord, il faut créer un projet de plug-in de manière classique.
- Ensuite, procédez de la même manière que dans le **cas n°1**.
- A présent, il reste à faire en sorte que le **JAR** puisse être utilisé en dehors de son plug-in dédié. Pour cela, toujours dans le fichier **MANIFEST.MF**, sélectionnez l'onglet **Runtime**.
- Dans la zone **Exported Package**, cliquez sur **Add...** et ainsi vous devriez voir apparaître l'ensemble des packages du **JAR**.
- Sélectionnez les tous, puis cliquez sur **Ok**.

Vous pouvez contrôler le fait que les packages sont bien indiqués comme exportés, en sélectionnant l'onglet **MANIFEST.MF**, puis en contrôlant la présence d'un code similaire à ceci :

```
Export-Package: com.mon.composant.une.bibliotheque.biz,  
               com.mon.composant.une.bibliotheque.bean,  
               com.mon.composant.une.bibliotheque.implement,  
               com.mon.composant.une.bibliotheque.util
```

En ce qui concerne la solution consistant à faire appel à l'assistant d'Eclipse, celle-ci produira le même résultat. Elle n'a d'avantage que le fait d'être plus rapide et légèrement plus aisée.

Pour ce faire :

- Allez dans le menu **File > New > Other... > Plug-in Development**, puis sélectionnez **Plug-in from Existing JAR Archives**.
- Sur la page qui s'affiche, cliquez sur **Add...** ou **Add External...** (en fonction de votre contexte) afin de sélectionner le ou les **JAR** à transformer en plug-in, puis cliquez sur **Next**.
- Spécifiez le nom de votre plug-in. Pour cela, vous aurez avantage à respecter la convention habituelle du nommage de vos plug-ins & packages, avec éventuellement en suffixe, le package principal du **JAR** ajouté.

- Réglez les autres paramètres comme bon vous semblera, puis cliquez sur Finish.

A présent, vous pouvez faire référence à ce nouveau plug-in, comme à n'importe quel autre, et la bibliothèque que vous avez incluse dans celui-ci est désormais accessible aux autres plug-ins.

Cette pratique est ce que l'on peut appeler la "bundleisation d'un JAR".

 **Remarque** : Avant de vous lancer dans la bundleisation d'un JAR, vérifiez que celui-ci ne l'est pas déjà. En effet, vous l'aurez compris, la bundleisation d'une bibliothèque implique simplement de mettre un ou plusieurs JAR dans un nouveau JAR de type plug-in, tout en ajoutant les déclarations OSGi nécessaires, au sein du fichier MANIFEST.MF. De fait, un certain nombre de bibliothèques, comme par exemple celles de la fondation Apache, peuvent déjà avoir été bundleisées. Donc, le cas échéant et sauf besoins particuliers, il sera parfaitement inutile de bundleiser ceux-ci une deuxième fois. Ainsi, pour effectuer cette vérification préalable, il suffit d'observer le contenu du JAR concerné (en le dézipant), notamment au niveau de son fichier META-INF/MANIFEST.MF. Si celui-ci comporte des variables OSGi telles que celles commençant par "Bundle-", alors cela signifie que c'est un JAR effectivement bundleisé. Dès lors, vous pouvez l'utiliser comme n'importe quel autre plug-in. Si vous constatez qu'un JAR n'est pas bundleisé, alors peut-être sera-t-il intéressant de faire une recherche préalable, au cas où une version bundleisée serait disponible par ailleurs. Vous pouvez par exemple, consulter les sites comme  **Orbit** ou  **SpringSource**, lesquels proposent des versions bundleisées de JAR. Dans le cas contraire, vous n'aurez plus qu'à le faire vous-même, comme expliqué précédemment.

[Sommaire](#) > [La Plate-Forme Eclipse](#) > [SWT / JFACE](#)

Définition de SWT et de JFace

Auteurs : Lunatix , djo.mos ,

SWT est une bibliothèque qui sert à construire des interfaces graphiques en Java, au même titre que les bibliothèques AWT et Swing.

La particularité de SWT est de déléguer l'affichage graphique au système d'exploitation. Concrètement, à travers SWT vous utilisez les MFC, Gtk ou autres.

JFace est une surcouche de SWT, davantage orientée objet, qui amène de plus un modèle MVC et des fonctionnalités supplémentaires.

Quelles sont les différences entre SWT et Swing ?

Auteurs : Lunatix ,

Swing est une API "lightweight", c'est-à-dire qu'elle redessine tous les composants graphiques (widgets) à l'aide des primitives de Java2D.

SWT quant à lui, utilise les API graphiques du système d'exploitation, c'est-à-dire que ses widgets sont dessinés directement par ce dernier.

En d'autres termes, pour afficher un bouton, Swing va dessiner un rectangle et placer un label au milieu, alors que SWT va simplement demander au système d'exploitation de se charger de cette opération.

Sous SWT, chaque objet Java représentant un élément graphique, n'est qu'un wrapper vers du code natif.

Quels sont les avantages et inconvénients de SWT par rapport à Swing ?

Auteurs : Lunatix ,

Ici l'idée est de simplement présenter quelques pistes de réflexion, afin de vous aider à faire votre choix. Prenez tout de même le temps d'approfondir le sujet, en vérifiant par exemple si ce qui est dit ici reste valable dans le temps.

Avantages

- **Non standard :** vous pouvez distribuer vous même SWT avec votre application ce qui permet de gérer finement le versioning, de bénéficier rapidement des BugFixs et des mises à jour
- **Natif :** SWT s'adapte automatiquement au Look and Feel du système d'exploitation. Une application Java sous SWT va s'adapter automatiquement à un windows customisé à l'extrême, ou à un Gtk avec un thème exotique
- **Petite API :** SWT est une petite API, avec un nombre limité de widgets, et peu d'héritage, ce qui en fait une API facile à prendre en main
- **Consommation mémoire :** la consommation mémoire de SWT est plus faible que celle de Swing
- **Vous pouvez bénéficier du projet Eclipse RCP** comme framework de développement
- **Open Source**
- **Linkable :** si vous prévoyez de compiler votre application avec GCJ et Gnu-Classpath, il est possible d'utiliser SWT facilement (il existe une version de Eclipse native)
- **Bonne intégration :** possibilité d'utiliser COM, des plug-ins, le browser web natif pour rendre du HTML

Inconvénients

- **Non standard** : vous devez distribuer vous même SWT avec votre application, ce qui va alourdir votre distribution et parfois compliquer le déploiement
- **Non standard** : moins de compétences sur SWT, moins de doc, moins d'aide sur le forum de DVP
- **Natif** : on subit les limitations de la plate-forme sous-jacente (taille maximum d'un champ texte, nombre d'image, pas de centrage des champs header, des tables, ...)
- **Gestion mémoire** : il est nécessaire de faire un `.dispose()` sur tous les éléments graphiques pour libérer la mémoire du composant natif qu'il représente
- **Petite API** : SWT/Jface est une petite API, et il manque des choses comparé à Swing
- **Portabilité** : Comparée à Swing et à la version Windows, la version Gtk de SWT reste lente ( **plus d'info**)
- **Pas d'applet possible**
- **Pas de SandBox possible** dans Java Web Start, à cause des accès natifs

Quels sont les Widgets SWT ?**Auteurs** : Lunatix ,

Vous pouvez voir les widgets SWT sur cette page :  **Widgets SWT**.
Chaque widget est présenté avec un exemple et sa Javadoc.

Où trouver des exemples sur SWT ?**Auteurs** : Lunatix ,

L'équipe de développement de SWT tient à jour une page d'exemples  **SWT Snippets**.
Vous pourrez y trouver de nombreux exemples d'utilisation de chacun des widgets SWT, ainsi que la réponse à des questions plus complexes, comme comment lancer une tâche longue dans un  **Thread**, gérer le  **Drag and Drop**, ...

[3.0-] Comment lancer une application SWT ?**Auteurs** : Lunatix ,

Pour lancer une application SWT, vous devez bien sûr avoir les jar SWT dans le classpath, mais aussi déclarer les librairies natives. Celles-ci se trouvent dans le sous-répertoire de plug-ins `org.eclipse.swt.[OS]`

- **Sous Eclipse 3.1M5+** : menu Run > Application SWT
- **Sous Eclipse** : menu Run > Run Java Application > New > Arguments > VM Arguments et ajouter - `Djava.library.path=c:\eclipse\plugins\org.eclipse.swt.win32_2.1.0\os\win32\x86`
- **Enfin**, pour le distribuer vous devez ajouter dans la ligne de commande - `Djava.library.path=chemin_vers_librairies`

Bien sûr, vous devrez adapter ces lignes de commandes à votre OS et à la version de SWT utilisée.

[3.2+] Comment lancer une application SWT ?

Auteurs : [djo.mos](#) ,

Dans les versions récentes d'Eclipse, pour lancer une application SWT, il suffit juste d'ajouter le jar de SWT correspondant à votre architecture (org.eclipse.swt.gtk.linux.x86_3.4.0.xxx.jar par exemple pour Linux/x86) Ce jar peut être récupéré du dossier plug-ins de votre installation Eclipse.

Comment inclure du Swing dans une application SWT et inversement ?

Auteurs : [Lunatic](#) , [Laurent.B](#) , [Gueritarish](#) ,

Pour inclure du Swing dans du SWT, voici un exemple [Snippet154.java](#).

Pour inclure du SWT dans du Swing, c'est cet exemple qu'il faut observer [Snippet337.java](#).

Vous avez quelques autres exemples dans la partie Swing/AWT de cette page [SWT Snippets](#).

Pourquoi mon widget qui hérite d'un widget SWT plante à l'exécution ?

Auteurs : [Lunatic](#) , [Laurent.B](#) ,

Il est fortement déconseillé d'hériter d'un widget SWT.

En effet, même si SWT est une API Java, les objets que l'on utilise ne sont que des représentations du code natif d'un toolkit graphique. De fait, l'héritage n'est pas forcément supporté (Gtk est codé en C par exemple).

En conséquence, chaque widget (sauf bien sûr 'Composite') utilise un mécanisme pour bloquer l'héritage :

là une méthode `protected void checkSubclass() throws SWTException`. Cf. la  [Javadoc](#).

Peut-être vous posez-vous la question de savoir pourquoi le mot clé final n'a pas été utilisé dans ce genre de cas ?

Tout simplement parce que le système de check au runtime n'empêche pas l'héritage. Ce qui fait qu'en cas de nécessité, il suffit de surcharger cette méthode dans votre classe.

Cependant, en choisissant cette option, vous vous exposez à des problèmes de non portabilité car si votre widget hérité fonctionne sur une plate-forme, il peut ne pas fonctionner sur une autre. Donc, à vous de savoir si ça vaut le coup de passer par l'héritage.

L'utilisation de ce mécanisme est à réserver pour faire du BugFix, sans attendre que cela soit intégré à la distribution officielle de SWT. Il ne faut pas s'en servir en revanche pour ajouter des fonctionnalités.

En clair, pas d'héritage de widget en SWT. La solution est d'utiliser la composition.

Comment packager une application SWT pour Java Web Start ?

Auteurs : [Lunatic](#) ,

La présence des bibliothèques natives de SWT impose de signer vos jar afin de pouvoir sortir de la Sandbox de Java Web Start.

Pour cela, il faut préparer un jar de votre application (myApp.jar), un jar avec les librairies natives (swt_win32.dll sous Windows) que l'on va nommer swt-native.jar et avoir sous la main swt.jar.

- 1 Pour générer votre propre certificat, en ligne de commande tapez la commande : `keytool -genkey -keystore myKeyStore -alias myKey`
où
- myKeyStore est le nom que vous voulez pour votre keyStore
- myKey est le nom que vous voulez pour votre clé
Répondez aux questions en n'oubliant pas le mot de passe et repondez yes à la dernière question.
- 2 Ensuite, tapez la commande : `keytool -selfcert -alias myKey -keystore myKeyStore` (attention au mot de passe)
- 3 Et pour finir
`jarsigner.exe -keystore myKeyStore -verbose -certs swt.jar myKey`
`jarsigner.exe -keystore myKeyStore -verbose -certs swt-native.jar myKey`
`jarsigner.exe -keystore myKeyStore -verbose -certs myApp.jar myKey`
- 4 Faire également de même avec toutes vos librairies (jface.jar, ...).

Puis comme fichier .jnlp

```
<?xml version="1.0" encoding="utf-8"?>
<jnlp spec="1.0+" codebase="http://www.domaine.com/application" href="myApp.jnlp">
  <information>
    <title>myApp</title>
    <vendor>Lunatix corporation inc</vendor>
    <homepage href="http://www.domaine.com/" />
    <description>myApp</description>
    <description kind="short">une application de test</description>
  </information>
  <security>
    <all-permissions/>
  </security>

  <resources>
    <j2se version="1.4+" />
    <jar href="myApp.jar" />
    <jar href="jface.jar" />
    <jar href="forms.jar" />
    <nativelib href="swt-native.jar" />
  </resources>

  <resources os="Windows">
    <jar href="swt.jar" />
  </resources>

  <application-desc main-class="com.domaine.myApp.MyApp" />
</jnlp>
```

Pour une version plus détaillée, lisez l'article  lfe.developpez.com/Java/SWT/WebStart/ de LFE.

[2.x] Comment afficher un calendrier ou un sélecteur de date en SWT ?

Auteurs : Lunatix ,

Il n'y a pas de calendrier ni de widget dédié aux dates dans SWT. Il faut donc utiliser des librairies supplémentaires

 SWTCalendar

 SWT DatePicker

[3.3+] Comment afficher un calendrier ou un sélecteur de date en SWT ?

Auteurs : djo.mos ,

Depuis la version 3.3, un composant de sélection de date/temps a été intégré au jeu de composants standard de SWT. C'est le widget  **DateTime**

Voici un exemple de son utilisation :

```
DateTime calendar = new DateTime (shell, SWT.CALENDAR);
DateTime time = new DateTime (shell, SWT.TIME);
```

Comment empêcher une fenêtre (Shell) d'être disposée à sa fermeture ?

Auteurs : Mickael Baron ,

Suite à un clic sur la croix de fermeture d'une fenêtre (Shell), celle-ci est "disposée" et par conséquent fermée. La fenêtre n'est pas simplement rendue invisible, ses ressources sont effectivement libérées, via un appel implicite de la méthode dispose().

Pour empêcher de "disposer" une fenêtre à sa fermeture, il suffit de placer un écouteur sur l'événement ShellEvent et traiter la méthode shellClosed(). L'événement ShellEvent véhicule un flag appelé *doit* dont le rôle est d'indiquer si l'opération en cours (fermeture, iconification, ...) doit être achevée ou pas. Dans notre cas, il suffit de mettre le flag à faux.

Par ailleurs, on rend la fenêtre Shell invisible en appelant sa méthode setVisible() avec *false* et on préserve ainsi l'instance de la fenêtre, pour notamment la faire réapparaître lorsque nécessaire.

```
shell.addShellListener(new ShellAdapter() {
/**
 * @see org.eclipse.swt.events.ShellAdapter#shellClosed(org.eclipse.swt.events.ShellEvent)
 */
@Override
public void shellClosed(ShellEvent event) {
// On neutralise l'événement
event.doit = false;
// On rend la fenêtre invisible
((Shell) event.getSource()).setVisible(false);
}
});
```

Sommaire > Gestion de version**[3.4+] Comment utiliser Subversion (SVN) dans Eclipse ?****Auteurs : djo.mos , Laurent.B ,****Deux plug-ins existent Subclipse et Subversive.****Le premier est de la société Tigris (éditeur de Subversion) et le second est de la société Polarion.****En plus d'évoluer sans cesse, ces plug-ins ont chacun leurs particularités, ce qui les rend difficiles à départager.****Le mieux étant de se faire sa propre opinion en les testant tous les deux, tout en évitant de les installer en même temps.****Ces deux plug-ins s'installent de façon classique à l'aide du Gestionnaire de Mises à Jour (Cf. Installation d'un plug-in avec [3.4] Comment installer un plug-in ? ou [3.5] Comment installer un plug-in ? si besoin).****Pour installer Subversive, voir la Q/R [3.4+] Comment installer le client SVN Subversive ?****Pour installer Subclipse, voir la Q/R [3.2+] Comment installer le client SVN Subclipse ?****[3.4+] Comment installer le client SVN Subversive ?****Auteurs : djo.mos , Laurent.B ,****Depuis la version 3.4, Subversive fait désormais partie des projets Eclipse.****Il est composé de deux parties, la première permet d'utiliser les différentes fonctionnalités SVN et la seconde permet de connecter Eclipse à un serveur SVN. Cette dernière étant toujours maintenue directement par Polarion (Cf. <http://community.polarion.com/index.php?page=overview&project=subversive> pour de plus amples renseignements).****1^{ère} étape : Si nécessaire, ajoutez les update-sites des deux parties de Subversive, via le Gestionnaire de Mises à Jour d'Eclipse (cf. Installation d'un plug-in dans Eclipse [3.4] Comment installer un plug-in ?, [3.5] Comment installer un plug-in ?, [3.6+] Comment installer un plug-in ? pour plus de détails).**

- **Eclipse 3.6 et supérieures :**
 - **L'environnement : Celui-ci est déjà référencé par défaut, dans le menu Help > Install New Software..., en sélectionnant l'update-site qui correspond à votre version :**
 - **Juno - <http://download.eclipse.org/release/juno>, dans la rubrique Collaboration Tools**
 - **Indigo - <http://download.eclipse.org/release/indigo>, dans la rubrique Collaboration Tools**
 - **Helios - <http://download.eclipse.org/release/helios>, dans la rubrique Collaboration Tools**
 - **Les connecteurs : L'installation d'un ou plusieurs connecteurs est en principe proposée ci-tôt l'environnement installé et après avoir redémarré Eclipse. Ceci permet en principe de se passer de l'ajout des update-sites des connecteurs mais si ce n'est pas le cas, voici malgré tout les update-sites en fonction des versions d'Eclipse :**
 - **Juno : <http://community.polarion.com/projects/subversive/download/integrations/juno-site/>**
 - **Indigo : <http://community.polarion.com/projects/subversive/download/eclipse/3.0/update-site/>**
 - **Helios : <http://community.polarion.com/projects/subversive/download/integrations/helios-site/>**

Si vous rencontrez des problèmes pour obtenir les connecteurs, consultez cette page

 [Download Subversive Subversion Team Provider for Eclipse](#)

- **Eclipse 3.5 :**
 - L'environnement : Celui-ci est déjà référencé par défaut, dans le menu Help > Install New Software..., en sélectionnant Galileo - <http://download.eclipse.org/release/galileo>, puis en ouvrant la rubrique Collaboration Tools
 - Les connecteurs : <http://community.polarion.com/projects/subversive/download/eclipse/2.0/galileo-site/>
- **Eclipse 3.4 :**
 - L'environnement : <http://download.eclipse.org/technology/subversive/0.7/update-site/>
 - Les connecteurs : <http://community.polarion.com/projects/subversive/download/eclipse/2.0/update-site/>

2^{ème} étape : Sélectionner les modules réellement nécessaires parmi tous ceux proposés.

- Le premier module à sélectionner est celui qui s'intitule Subversive SVN Team Provider (incubation).
- Le deuxième module à sélectionner est un des connecteurs disponibles, lequel doit être adapté au serveur SVN que vous utilisez, ainsi qu'à votre environnement de développement.
Si vous ignorez lequel sélectionner, le plus simple étant de cocher tous les connecteurs présents dans la rubrique Subversive SVN Connectors car de toute manière, il est ensuite possible de sélectionner le connecteur le mieux adapté, au niveau des préférences d'Eclipse, dans le menu Team > SVN et l'onglet SVN Connector.

Une fois les plug-ins sélectionnés, il suffit de valider l'installation et laisser Eclipse redémarrer.

Remarque : Les update-sites que vous venez d'ajouter proposent différents modules complémentaires, tels que ceux qui contiennent le code source de Subversive ou ceux qui permettent son intégration dans d'autres plug-ins comme Mylyn, Buckminster et ProjectSet. Si vous vous demandez à quoi ils peuvent servir, c'est probablement que vous pouvez vous abstenir de les installer car ils ne vous seront pas utiles.

Si votre version de Eclipse est antérieure à la version 3.4, il reste possible d'installer le plug-in Subversive.

Pour cela, reportez-vous au lien suivant  <http://community.polarion.com/index.php?page=download&project=subversive>

[3.2+] Comment installer le client SVN Subclipse ?

Auteurs : [djo.mos](#) , [Laurent.B](#) ,

1^{ère} étape : Ajouter un des update-sites de Subclipse, via le Gestionnaire de Mises à Jour d'Eclipse (si besoin, cf. Installation d'un plug-in dans Eclipse [3.4] Comment installer un plug-in ?, [3.5] Comment installer un plug-in ?, [3.6+] Comment installer un plug-in ?).

Les versions d'Eclipse 3.2 à 4.2 partagent les mêmes URL en ce qui concerne les update-sites à ajouter pour Subclipse.

En revanche, il existe plusieurs update-sites selon la version de SVN ciblée et donc du plug-in Subclipse convenant à celle-ci. A vous d'utiliser celle qui convient le mieux par rapport au serveur SVN auquel vous souhaitez connecter vos projets.

- Subclipse 1.8.x (pour SVN 1.7.x) : http://subclipse.tigris.org/update_1.8.x
- Subclipse 1.6.x (pour SVN 1.6.x) : http://subclipse.tigris.org/update_1.6.x
- Subclipse 1.4.x (pour SVN 1.5.x) : http://subclipse.tigris.org/update_1.4.x

2^{ème} étape : Sélectionner les modules nécessaires parmi ceux proposés.

- Après avoir ajouté un des update-sites, une liste de plusieurs plug-ins apparaît.
A moins de savoir précisément ce dont vous avez besoin, par simplicité sélectionnez-les tous.

Une fois les plug-ins sélectionnés, il suffit de valider l'installation et laisser Eclipse redémarrer.

Remarque : Pour fonctionner, Subclipse requiert, soit le module JavaHL (Java High Level) lequel s'appuie sur les bibliothèques natives du client SVN installé sur le poste, soit le module SVNKit qui lui est codé en Java et n'a pas besoin de client SVN préinstallé. Donc vous l'aurez compris, si vous n'avez pas installé préalablement de client natif pour utiliser SVN, que ce soit le client en ligne de commande ou le client graphique tel que Tortoise, vous serez contraint d'utiliser le module SVNKit car le module JavaHL ne fonctionnera pas. Tigris dit s'efforcer de maintenir les API natives et SVNKit au même niveau, donc que Subclipse fonctionne à l'aide de l'un ou de l'autre, le niveau des fonctionnalités devrait théoriquement être le même.

Le choix entre les deux modes se paramètre dans les préférences d'Eclipse, via le menu Window > Preferences > Team > SVN, au niveau du bloc SVN Interface, dans lequel vous pouvez sélectionner le client JavaHL (JNI) ou le client SVNKit (pure Java).

Les liens suivants pourront vous être utiles pour de plus amples renseignements :

- Référentiel des update-sites de Subclipse  <http://subclipse.tigris.org/servlets/ProjectProcess?pagelD=p4wYuA>.
- Les particularités concernant Windows 64 bits, OS X et Linux, ainsi que diverses autres explications  <http://subclipse.tigris.org/wiki/JavaHL>.

Comment configurer Eclipse pour fonctionner avec CVS ?

Auteurs : Lunatix ,

Ouvrir la perspective CVS, puis dans la vue CVS repositories > bouton droit > new repository location > ouverture de l'assistant de connexion.

Une fois que votre repository est bien renseigné, vous pourrez naviguer dedans dans l'arborescence.

Comment extraire un projet sous CVS ?

Auteurs : Lunatix ,

Ouvrez la perspective CVS, puis dans votre repository CVS, choisissez ce que vous voulez pour le checkout. faites bouton droit > check out as > java project

Comment changer un projet de repository CVS ?

Auteurs : Lunatix ,

Pour changer la localisation de votre cvs, allez dans les propriétés du projet : CVS > change sharing > choisissez un autre repository

Ce deuxième serveur CVS doit déjà être configuré dans la perspective CVS, et doit être compatible avec votre projet.

Autres systèmes de gestion de version que CVS ?

Auteurs : Lunatix , Laurent.B ,

Pour gérer d'autres gestionnaires de version que CVS, il vous faudra installer des plug-ins supplémentaires.

- Subversion :  Subclipse
- ClearCase :  IBM Rational ClearCase plug-ins
- VSS :  VSS Plugin
- Git :  EGit (dans les versions récentes d'Eclipse et selon le package choisi, ce plugin peut déjà être installé par défaut)

Utiliser le gestionnaire de version intégré ?

Auteurs : Lunatix , Laurent.B ,

Eclipse, en plus de permettre l'utilisation de gestionnaires de version tels que CVS, SVN, Git... celui-ci dispose également d'un dépôt de version interne.

Pour y accéder, il suffit de faire, dans l'éditeur ou dans le Package Explorer, un clic droit sur le fichier concerné, puis de sélectionner le menu Compare With > Local History.... Ainsi vous pourrez visualiser les différences avec des versions précédentes de votre fichier.

A noter qu'il est également possible de remplacer la version courante d'un fichier source, par une version issue de l'historique, en sélectionnant à la place le menu Replace With > Local History....

Remarque : afin d'éviter quelques surprises telles que l'absence d'historique pour un fichier en particulier ou encore, des versions non conservées car trop anciennes, il peut être intéressant de modifier quelques paramètres, au niveau du menu Window > Preferences > General > Workspace > Local History. Ainsi, on

peut indiquer la durée de conservation, le nombre de modifications à conserver et la taille des fichiers à prendre en compte.

A noter toutefois que tout ceci n'a rien de pérenne et ne remplacera pas un véritable gestionnaire de version.

Sommaire > Plug-ins

Qu'est-ce qu'un plug-in sous Eclipse ?

Auteurs : [djo.mos](#) ,

Basiquement, un plug-in au sens d'Eclipse est un ou plusieurs fichiers jar qui peuvent agir sur Eclipse en ajoutant de nouvelles fonctionnalités, perspectives, vues, etc.

Techniquement, un plug-in au sens d'Eclipse est un bundle OSGi qui peut agir sur Eclipse :

- En exportant des packages (au sens d'OSGi) ce qui permet à d'autres plug-ins d'utiliser les classes qu'il exporte.
- En utilisant des points d'extension définies par d'autres plug-ins.
- En exposant des points d'extension que d'autres plug-ins pourront utiliser.

Pour développer des plug-ins pour Eclipse, on utilise Eclipse lui-même, ou plus précisément le tooling PDE (Plug-in Development Tools).

Où trouver des plug-ins pour Eclipse ?

Auteurs : [Jfrancois57](#) ,

C/C++ IDE - Graphical Editor Framework (GEF) Project - COBOL - UML2  eclipse.org/tools/

Site généraliste  eclipse-plugins.2y.net

Site généraliste  eclipseplugincentral

[3.3-] Comment installer un plug-in ?

Auteurs : [Jfrancois57](#) , [djo.mos](#) , [Laurent.B](#) ,

Voici les différentes façons d'installer un plug-in dans Eclipse 3.3 et versions antérieures :

Lorsque le plug-in se présente sous la forme d'un fichier compressé (zip ou tar)

- Téléchargez le fichier
- Décompressez le fichier zip ou tar
- Mettre respectivement les fichiers /plugins dans le répertoire /eclipse/plugins et les fichiers /features dans /eclipse/features (s'il y a également des features)
- Redémarrez Eclipse

Lorsque le plug-in se présente sous la forme d'une adresse web (update site)

- Ouvrez Eclipse
- Allez dans le menu Help > Software Updates > Find and Install
- Sélectionnez Search for new features to install
- Cliquez sur New Remote Site, donnez lui un nom (il sera identifié ainsi dans la liste) et entrez l'URL
- Cliquez sur Next, et continuez jusqu'au bout du processus en acceptant toutes les demandes
- Eclipse devrait télécharger et installer automatiquement le plug-in après avoir cliqué sur Install. Le programme vous demande ensuite de redémarrer. Cliquez sur Yes

Lorsque le plug-in se présente sous la forme d'un update-site archivé

- Téléchargez le fichier
- Décompressez le fichier zip ou tar
- Ouvrez Eclipse
- Allez dans le menu Help > Software Updates > Find and Install
- Sélectionnez Search for new features to install
- Cliquez sur New Local Site et indiquez l'emplacement de l'archive extraite, puis donnez un nom pour l'update site
- Cliquez sur Next, et continuez jusqu'au bout du processus en acceptant toutes les demandes
- Eclipse devrait installer automatiquement le plug-in après avoir cliqué sur Install. Le programme vous demande ensuite de redémarrer. Cliquez sur Yes

[3.4] Comment installer un plug-in ?

Auteurs : Jfrancois57 , djo.mos , Laurent.B ,

Voici les différentes façons d'installer un plug-in dans Eclipse Ganymede (3.4) :

Lorsque le plug-in se présente sous la forme d'une adresse web, appelée "update-site"

- Ouvrez Eclipse
- Allez dans le menu Help > Software Updates puis dans la section Available Software
- Cliquez sur Add Site, entrez l'URL et validez
- Localisez le nouveau site dans la liste, déroulez-le et cochez les composants qui vous intéressent.
- Cliquez sur le bouton Install
- Eclipse installe alors automatiquement le(s) plug-in(s). Le programme vous demande ensuite de redémarrer. Cliquez sur Yes

Lorsque le plug-in se présente sous la forme d'un fichier compressé (zip ou tar)

- Téléchargez le fichier
- Décompressez le fichier zip ou tar
- Mettre respectivement les fichiers /plugins dans le répertoire /eclipse/plugins et les fichiers /features dans /eclipse/features (s'il y a également des features)
- Redémarrez Eclipse

Lorsque le plug-in se présente sous la forme d'un fichier compressé (zip) 2^{ème} méthode

- Téléchargez le fichier
- Déposer le fichier dans le répertoire dropins (présent à la racine du répertoire d'installation d'Eclipse)
- Démarrer ou redémarrer Eclipse

Lorsque le plug-in se présente sous la forme d'un update-site archivé

- Téléchargez le fichier
- Ouvrez Eclipse
- Allez dans le menu Help > Software Updates puis dans la section Available Software
- Cliquez sur Add Site, puis sur Archive. Localisez l'archive et validez
- Cliquez sur le bouton Install

- Eclipse installe alors automatiquement le(s) plug-in(s). Le programme vous demande ensuite de redémarrer. Cliquez sur Yes

[3.5] Comment installer un plug-in ?

Auteurs : Jfrancois57 , djo.mos , Laurent.B ,

Voici les différentes façons d'installer un plug-in dans Eclipse Galileo (3.5) :

Lorsque le plug-in se présente sous la forme d'une adresse web, appelée "update-site"

- Ouvrez Eclipse
- Allez dans le menu Help > Install New Software... puis cliquez sur le bouton Add...
- Dans la fenêtre qui s'est ouverte, laissez vide ou donnez le nom que vous voulez à l'update-site que vous êtes en train d'ajouter, renseignez son URL, puis validez.
- L'update-site que vous venez d'ajouter se retrouve automatiquement sélectionné dans la liste "Work with". Vous pouvez donc cocher les différents plug-ins qui vous intéressent parmi la liste proposée.
- Cliquez sur Next une première fois, puis une deuxième si aucune erreur de dépendance n'est signalée.
- Acceptez les licences, puis cliquez sur Finish.
- Comme recommandé, redémarrez Eclipse.

Lorsque le plug-in se présente sous la forme d'un fichier compressé (zip ou tar)

- Téléchargez le fichier
- Décompressez le fichier zip ou tar
- Mettre respectivement les fichiers /plugins dans le répertoire /eclipse/plugins et les fichiers /features dans /eclipse/features (s'il y a également des features)
- Redémarrez Eclipse

Lorsque le plug-in se présente sous la forme d'un fichier compressé (zip) 2^{ème} méthode

- Téléchargez le fichier
- Déposer le fichier dans le répertoire dropins (présent à la racine du répertoire d'installation d'Eclipse)
- Démarrer ou redémarrer Eclipse

Lorsque le plug-in se présente sous la forme d'un update-site archivé

- Téléchargez le fichier
- Ouvrez Eclipse
- Allez dans le menu Help > Install New Software... puis cliquez sur le bouton Add...
- Dans la fenêtre qui s'est ouverte, laissez vide ou donnez le nom que vous voulez à l'update-site que vous êtes en train d'ajouter.
- Cliquez sur Archive, puis localisez l'archive et validez
- L'update-site que vous venez d'ajouter se retrouve automatiquement sélectionné dans la liste "Work with". Vous pouvez donc cocher les différents plug-ins qui vous intéressent parmi la liste proposée.
- Cliquez sur Next une première fois, puis une deuxième si aucune erreur de dépendance n'est signalée.
- Acceptez les licences, puis cliquez sur Finish.

- Comme recommandé, redémarrez Eclipse.

[3.6+] Comment installer un plug-in ?

Auteurs : Laurent.B ,

La façon de faire avec la version d'Eclipse 3.5 reste valable pour tous les plug-ins (Cf. [3.5] Comment installer un plug-in ?).

Cependant, depuis Eclipse 3.6, il existe un nouveau moyen, lequel consiste à passer par l'Eclipse Marketplace.

Cela concerne les plug-ins et outils de sociétés ou contributeurs tiers, que l'on peut alors installer directement à partir d'Eclipse.

Donc, à la manière de ce qui se fait dans d'autres logiciels, il est à présent possible de rechercher un plug-in ou un outil puis une fois trouvé, de l'installer par un simple clic.

Par exemple, si vous voulez installer le plug-in pour Maven 2, allez dans le menu Help > Eclipse Marketplace..., puis saisissez maven 2 dans la case Find de la fenêtre qui s'est ouverte.

Validez en appuyant sur Entrée et en principe, le premier résultat devrait correspondre à Maven 2.

Il vous suffit alors de cliquer sur son bouton Install et de suivre les instructions.

Veillez noter que tous les plug-ins du Marketplace ne sont pas forcément installables directement.

Parfois vous n'aurez pas le bouton Install et il faudra aller sur le site de l'éditeur pour pouvoir le récupérer.

[3.2-] Comment configurer le proxy dans Eclipse ?

Auteurs : Jfrancois57 , Laurent.B ,

Si votre configuration réseau le nécessite, vous avez la possibilité de configurer un proxy afin qu'Eclipse puisse accéder à Internet.

La configuration s'effectue à partir du menu :

Window > Preferences > Install-Update, cocher la case "Enable HTTP Proxy connection" puis, compléter les paramètres "HTTP proxy host address" et "HTTP proxy host port".

[3.3+] Comment configurer le proxy dans Eclipse ?

Auteurs : Jfrancois57 , Laurent.B ,

Si votre configuration réseau le nécessite, vous avez la possibilité de configurer un proxy afin qu'Eclipse puisse accéder à Internet.

La configuration s'effectue à partir du menu :

Window > Preferences > General > Network Connections >, cocher la case "Manual proxy configuration" puis, compléter l'adresse "HTTP Proxy" et le numéro de "Port". A noter que les versions récentes d'Eclipse

offrent un certain nombre de paramètres supplémentaires tels que la configuration SSL / SSH2 / SOCKS, l'authentification, ...etc.

[2.x] Comment savoir si un plug-in est intégré/installé dans Eclipse ?

Auteurs : [zsoh](#) ,

Pour voir la liste des plug-ins : cliquez sur le menu > Aide > A propos et onglet Détails des plug-ins

[3.x] Comment savoir si un plug-in est intégré/installé dans Eclipse ?

Auteurs : [djo.mos](#) ,

Pour voir la liste des plug-ins : cliquez sur le menu > Help > About Eclipse Platform puis sur le bouton onglet Plug-in Details

Pourquoi mon plug-in n'apparaît pas dans les perspectives ?

Auteurs : [L'équipe Eclipse](#) ,

Il arrive que la détection d'un nouveau plug-in se passe mal. Dans ce cas, il vous suffit de redémarrer Eclipse avec l'option -clean pour forcer la détection des plug-ins. Ne laissez pas cette option, car cela ralentit considérablement le démarrage.

Quels sont les langages que peut gérer Eclipse en plus de Java ?

Auteurs : [Lunatix](#) , [djo.mos](#) , [Laurent.B](#) ,

Grâce à son architecture modulaire et à l'ajout de plug-ins complémentaires, Eclipse peut gérer un grand nombre de langages. Cette liste n'est pas exhaustive et tous les plug-ins ne sont pas de qualité équivalente.

- C/C++ : Plug-in officiel :  www.eclipse.org/cdt/
- Cobol : Plug-in officiel :  www.eclipse.org/cobol/
- DLTK : Plug-in officiel qui offre le support de Tcl, Ruby et Python :  <http://www.eclipse.org/dltk/>
- Php : Plug-in officiel PDT  www.eclipse.org/pdt/ ou PHPEclipse  www.phpeclipse.de
- Python :  [pydev](http://pydev.org/)
- Latex :  [Texlipse](http://www.texlipse.com/)
- Perl :  e-p-i-c.sourceforge.net/
- C# :  [improve-technologies](http://www.improve-technologies.com/)  [esharp](http://www.esharp.com/),  [Emonic](http://www.emonic.com/)

- Ruby :  [rubeclipse](#)

Comment faire une interface graphique en Java sous Eclipse ?

Auteurs : [Lunatic](#) , [djo.mos](#) , [Laurent.B](#) ,

Depuis la version Indigo (3.7), Eclipse dispose enfin d'un éditeur graphique (WYSIWYG) pour la création d'IHM développées en Java. Ce plug-in s'appelle WindowBuilder Pro (cf. [FAQ Comment installer WindowBuilder Pro](#) ?).

Pour les versions antérieures à Eclipse 3.7, ce plug-in fonctionne également avec les versions 3.4, 3.5, et 3.6.

Avant WindowBuilder, il existait d'autres possibilités mais moins abouties ou payantes dans le cadre d'une utilisation commerciale :

-  **Visual Editor Project (VEP)** : le plug-in officiel qui précédait WindowBuilder. Gratuit, gérant Swing et SWT mais qui n'a pas suivi les versions successives d'Eclipse.
-  **Visual Swing for Eclipse** gérant Swing.
-  **Jigloo** Gratuit pour usage non professionnel, gérant Swing et SWT.

Quoi qu'il en soit,  **Window Builder Pro** est à présent la référence puisqu'il est officialisé parmi les plug-ins d'Eclipse et que de plus, il est plutôt complet car il gère à la fois Swing, SWT, RCP et XWT.

Sa documentation est disponible dans l'aide Eclipse  [WindowBuilder User Guide](#)

Comment installer le Visual Editor (VE) ?

Auteurs : [Lunatic](#) , [djo.mos](#) , [Laurent.B](#) ,

 **Visual Editor est un plug-in qui a été abandonné depuis longtemps, puis finalement archivé et n'est pas compatible avec les versions récentes d'Eclipse.**

L'utilisation de Visual Editor n'a plus aucun intérêt depuis l'arrivée de  Window Builder Pro, qui de plus le surpasse largement.

 **Visual Editor (VE)** a été le plug-in officiel pour le développement d'interfaces graphiques Java d'Eclipse, avant l'arrivée de WindowBuilder.

Dans les versions d'Eclipse < 3.3, celui-ci s'installait de la manière suivante :

- 1 Aller dans le menu Help > Software Updates > Find and Install > Search for new features to install
- 2 Deployer Eclipse.org update site
- 3 Sélectionner EMF, GEF, and Visual Editor update sites
- 4 Choisir Visual Editor
- 5 Eclipse indique alors les versions de EMF et GEF compatibles. Il ne reste plus qu'à accepter les licenses et valider l'installation.

Officiellement VE n'était installable que sur Eclipse 3.2 (Callisto) mais qu'il était possible de le faire fonctionner dans :

- Eclipse 3.1.2 (procédure plus disponible)
- Eclipse 3.3 et 3.4 en suivant  [cette procédure](#)

Comment ajouter un composant au Visual Editor (VE) ?

Auteurs : [guipom](#) , [Laurent.B](#) ,

Informations et Liens obsolètes et inexploitable ...

http://dev.eclipse.org/viewcvvs/indextools.cgi/%7Echeckout%7E/org.eclipse.ve.examples/org.eclipse.ve.example.customwidget/WebContent/index_files/main.htm "Extending The Visual Editor Tutorial: Enabling support for a custom widget"

Visual Editor est un plug-in qui a été abandonné depuis longtemps, puis finalement archivé et n'est pas compatible avec les versions récentes d'Eclipse.

L'utilisation de Visual Editor n'a plus aucun intérêt depuis l'arrivée de  [Window Builder Pro](#), qui de plus le surpasse largement.

Comment installer WindowBuilder Pro ?

Auteurs : [Laurent.B](#) ,

WindowBuilder est présent par défaut mais uniquement dans le package  [Eclipse IDE for Java Developers](#) à partir d'Eclipse Indigo (3.7).

Si vous avez bien une version d'Eclipse supérieure ou égale à la 3.7 mais que ce n'est pas votre package de départ, vous pouvez néanmoins installer WindowBuilder en utilisant le gestionnaire de mises à jour, de la manière suivante :

- 1 Aller dans le menu Help > Install New Software...
- 2 Sélectionner l'update-site Indigo - <http://download.eclipse.org/releases/indigo> ou Juno - <http://download.eclipse.org/releases/juno> selon le cas, puis la catégorie General Purpose Tools
- 3 Sélectionner les éléments relatifs à WindowBuilder au sein de la liste.
- 4 Cliquer sur Next puis suivre les instructions pour terminer l'installation.

Pour les versions antérieures à la 3.7 ou même au cas où, voici les URL de tous les update-site en fonction des versions courantes :

- Eclipse 4.2 : <http://download.eclipse.org/windowbuilder/WB/release/R201206261200/4.2/>
- Eclipse 3.8 : <http://download.eclipse.org/windowbuilder/WB/release/R201206261200/3.8/>
- Eclipse 3.7 : <http://download.eclipse.org/windowbuilder/WB/release/R201206261200/3.7/>
- Eclipse 3.6 : <http://download.eclipse.org/windowbuilder/WB/release/R201206261200/3.6/>
- Eclipse 3.5 : <http://download.eclipse.org/windowbuilder/WB/release/R201206261200/3.5/>
- Eclipse 3.4 : <http://download.eclipse.org/windowbuilder/WB/release/R201106211200/3.4/>

Si vous constatez qu'une URL n'est plus valide, ou si vous souhaitez obtenir des versions zippées, vous pouvez vous référer à cette page :  [Installing WindowBuilder Pro](#)