

 eBook Gratuit

# APPRENEZ eclipse

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#eclipse

# Table des matières

À propos.....	1
<b>Chapitre 1: Se lancer dans l'éclipse.....</b>	<b>2</b>
Remarques.....	2
Versions.....	2
Exemples.....	2
Installation et configuration.....	3
Installer Marketplace dans Eclipse.....	4
Raccourcis clavier utiles.....	5
<b>Gérer des fichiers et des projets.....</b>	<b>5</b>
<b>Fenêtre de l'éditeur.....</b>	<b>5</b>
<b>Naviguer dans l'éditeur.....</b>	<b>5</b>
<b>Éditer le texte.....</b>	<b>5</b>
<b>Rechercher et remplacer.....</b>	<b>6</b>
<b>Déplacer un bloc de code.....</b>	<b>6</b>
Créer et exécuter un programme Java HelloWorld.....	6
Créer un nouveau projet Java.....	6
Créer une nouvelle classe Java.....	8
Exécuter votre classe Java.....	10
Importation de projets existants.....	10
<b>Chapitre 2: Comment Eclipse Remote Debugging fonctionne en coulisse.....</b>	<b>13</b>
Exemples.....	13
Comment fonctionne Eclipse Remote Debugging derrière les scènes.....	13
<b>Chapitre 3: Configuration d'Eclipse.....</b>	<b>15</b>
Exemples.....	15
Augmentation de la mémoire de tas maximale pour Eclipse.....	15
Spécifier la JVM.....	15
Comment configurer la taille de police des vues dans Eclipse sous Linux.....	16
<b>Chapitre 4: Configurer Eclipse pour C ++.....</b>	<b>19</b>
Exemples.....	19

Linux + CMake (générateur "Makefiles Unix") + Qt (optionnel).....	19
Qt (optionnel).....	19
Espace de travail.....	19
Attacher des sources au projet.....	19
Générateur de CMake.....	19
Construire.....	20
Ré-exécuter CMake (pour générer à nouveau les makefiles).....	20
<b>Chapitre 5: Créer un nouvel espace de travail dans Eclipse.....</b>	<b>21</b>
Exemples.....	21
Comment créer un espace de travail.....	21
<b>Chapitre 6: Débogage à distance dans Eclipse.....</b>	<b>22</b>
Exemples.....	22
Configurer le débogage à distance Eclipse pour une application.....	22
<b>Chapitre 7: Débogage des programmes Java dans Eclipse.....</b>	<b>24</b>
Exemples.....	24
Évaluation d'expressions dans une session de débogage.....	24
Débogage à distance d'une application Java.....	26
<b>Chapitre 8: Procédure de déploiement Tomcat.....</b>	<b>28</b>
Exemples.....	28
Procédure quand rien d'autre n'aide.....	28
<b>Chapitre 9: Raccourcis Eclipse.....</b>	<b>30</b>
Introduction.....	30
Exemples.....	30
Code commentaire / décomment.....	30
Ouvrir le dialogue de recherche.....	30
Pour obtenir une impression.....	30
Générer des Getters et des Setters.....	30
Texte mis en évidence par le refactor.....	30
Format xml.....	30
<b>Crédits.....</b>	<b>32</b>

---

# À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [eclipse](#)

It is an unofficial and free eclipse ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official eclipse.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapitre 1: Se lancer dans l'éclipse

## Remarques

Cette section fournit une vue d'ensemble de ce qu'est une éclipse et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans éclipse, et établir un lien avec les sujets connexes. La documentation pour éclipse étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

## Versions

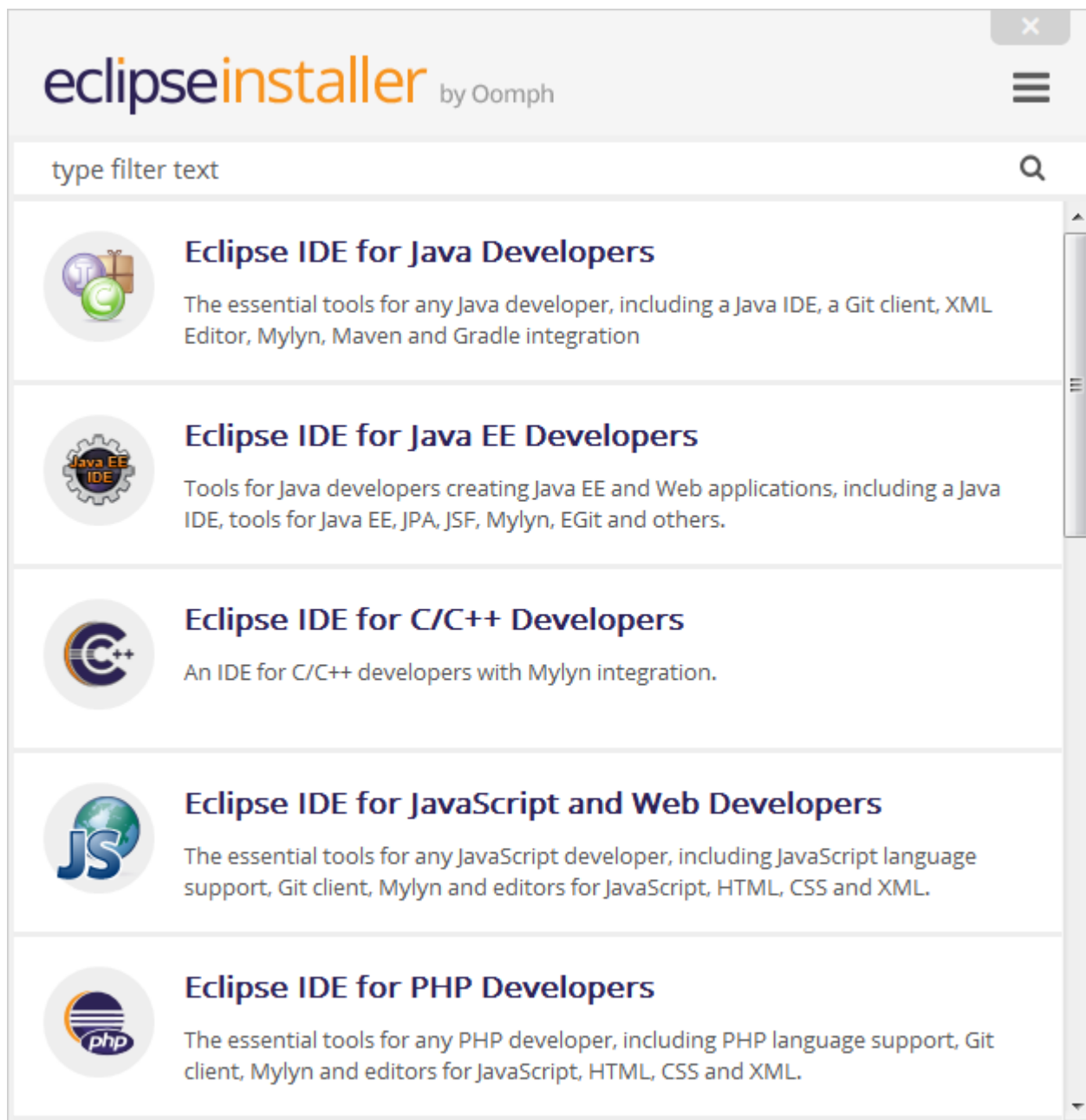
Version	prénom	Date de sortie
3.0		2004-06-21
3.1		2005-06-28
3.2	Callisto	2006-06-30
3.3	Europa	2007-06-29
3.4	Ganymède	2008-06-25
3.5	Galileo	2009-06-24
3.6	Helios	2010-06-23
3.7	Indigo	2011-06-22
3.8 et 4.2	Juno	2012-06-27
4.3	Kepler	2013-06-26
4.4	Luna	2014-06-25
4.5	Mars	2015-06-24
4.6	Néon	2016-06-22
4.7	Oxygène	2017-06-28
4.8	Photon (planifié)	2018-06-01

## Exemples

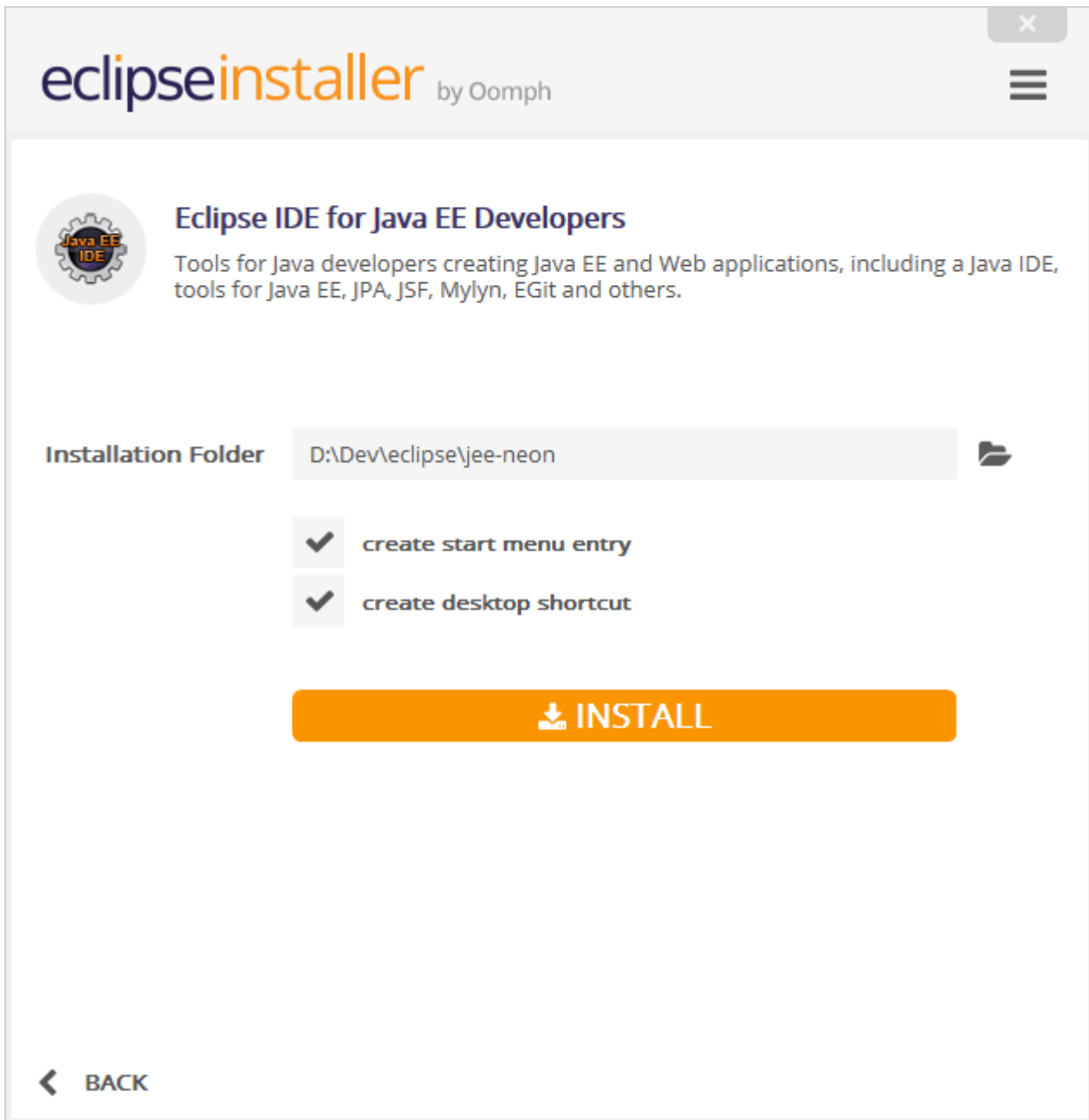
## Installation et configuration

Pour installer Eclipse, accédez à la page Web [Eclipse Downloads](#) où il existe généralement un lien direct pour télécharger la dernière version d'Eclipse. À partir d'Eclipse Mars (version 4.5), vous pouvez télécharger un programme d'installation qui vous guide tout au long de la procédure d'installation, au lieu de télécharger l'ensemble de l'installation en tant que fichier compressé (cette option est toutefois toujours disponible). Il existe également des liens pour télécharger les anciens packages Eclipse.

Eclipse se décline en plusieurs packages qui ciblent différents utilisateurs, comme illustré dans la capture d'écran ci-dessous de l'installateur. Par exemple, **l'IDE Eclipse pour les développeurs Java** contient des outils de base qui prennent en charge en développement, le débogage et la création d' applications Java, ainsi que le soutien de base pour le contrôle de version par exemple un plugin qui permet aux projets de versioning utilisant Git, tandis que **l'Eclipse pour les développeurs Android** fournit un environnement pour créer des applications Android.



Une fois qu'un paquet est sélectionné, la page suivante permet à l'utilisateur de sélectionner le répertoire d'installation, ainsi que d'autres options. La capture d'écran suivante illustre la procédure sur un ordinateur Windows.



Cliquez sur le bouton Installer pour lancer l'installation du package Eclipse dans ce répertoire.

Si Java est déjà correctement installé sur la machine, Eclipse doit être lancé correctement et aucune configuration ou configuration n'est généralement requise. Cependant, il est conseillé de modifier certaines options de configuration pour Eclipse, par exemple pour spécifier dans quelle JVM Eclipse doit être exécutée et pour configurer la mémoire minimale et maximale pour cette JVM. Pour ce faire, un fichier appelé `eclipse.ini` existe dans le répertoire d'installation, où se trouve cette configuration de démarrage. [Cette page](#) contient des détails sur la configuration de Eclipse à l'aide de ce fichier.

## Installer Marketplace dans Eclipse

Peu de versions classiques d'eclipse ne sont pas pré-installées avec marketplace, cela peut être installé en procédant comme suit:

1. Aller à l'aide → Installer un nouveau logiciel
2. Ajouter un nouveau référentiel (site spécifié ci-dessous)
3. Outils à usage général → Client de marché
4. Cliquez sur Terminer et vous avez terminé.

Sites de mise à jour du marché:

```
Oxygen - http://download.eclipse.org/releases/oxygen/  
Neon - http://download.eclipse.org/releases/neon/  
Mars - http://download.eclipse.org/releases/mars/  
Luna - http://download.eclipse.org/mpc/luna  
Helios - http://download.eclipse.org/releases/helios  
Juno - http://download.eclipse.org/releases/juno/
```

Raccourcis clavier utiles

---

## Gérer des fichiers et des projets

- **Ctrl + Shift + R** : Open Resource (fichier, dossier ou projet)
- **Ctrl + Shift + S** : Enregistrer tous les fichiers
- **Ctrl + W** : Fermer le fichier en cours
- **Ctrl + Shift + W** : Ferme tous les fichiers

---

## Fenêtre de l'éditeur

- **F12** : Passer à la fenêtre de l'éditeur
- **Ctrl + E** : Affiche la liste des éditeurs ouverts. Utilisez les touches fléchées et entrez pour changer
- **Ctrl + Page Down / Up** : Basculer vers l'éditeur / passer à l'éditeur précédent
- **Ctrl + M** : agrandit ou réduit la fenêtre actuelle de l'éditeur

---

## Naviguer dans l'éditeur

- **Ctrl + L** : aller à la ligne
- **Ctrl + Q** : Aller au dernier emplacement modifié
- **Ctrl + Shift + P** : avec une parenthèse sélectionnée: passer à la parenthèse de fermeture ou d'ouverture correspondante
- **Ctrl + Flèche vers le bas / Haut** : éditeur de défilement sans modification de la position du curseur



# Éditer le texte

- **Ctrl + D** : Supprimer la ligne
- **Alt + Maj + Y** : Enroulez le texte sélectionné (ajustez la largeur du texte à l'écran)
- **Alt + Shift + S** : options du menu Open Source
- **Alt + Maj + R** : mot mis en évidence par le refactor dans tous les fichiers
- **Ctrl + Alt + Haut / Ctrl + Alt + Bas** : Copie les lignes sélectionnées en haut / bas.

---

## Rechercher et remplacer

- **Ctrl + J** : Tapez un terme à rechercher, puis utilisez Ctrl + J / Ctrl + Maj + J pour remonter / reculer
- **Ctrl + K / Ctrl + Maj + K** : Ctrl + H puis fermez la fenêtre de recherche. Recherchez ensuite l'occurrence précédente / suivante du terme de recherche.

---

## Déplacer un bloc de code

- **Maj + tab** : Déplacer vers la gauche
- **Alt + Haut / Alt + Bas** : Déplacer vers le haut / bas.

### Créer et exécuter un programme Java HelloWorld

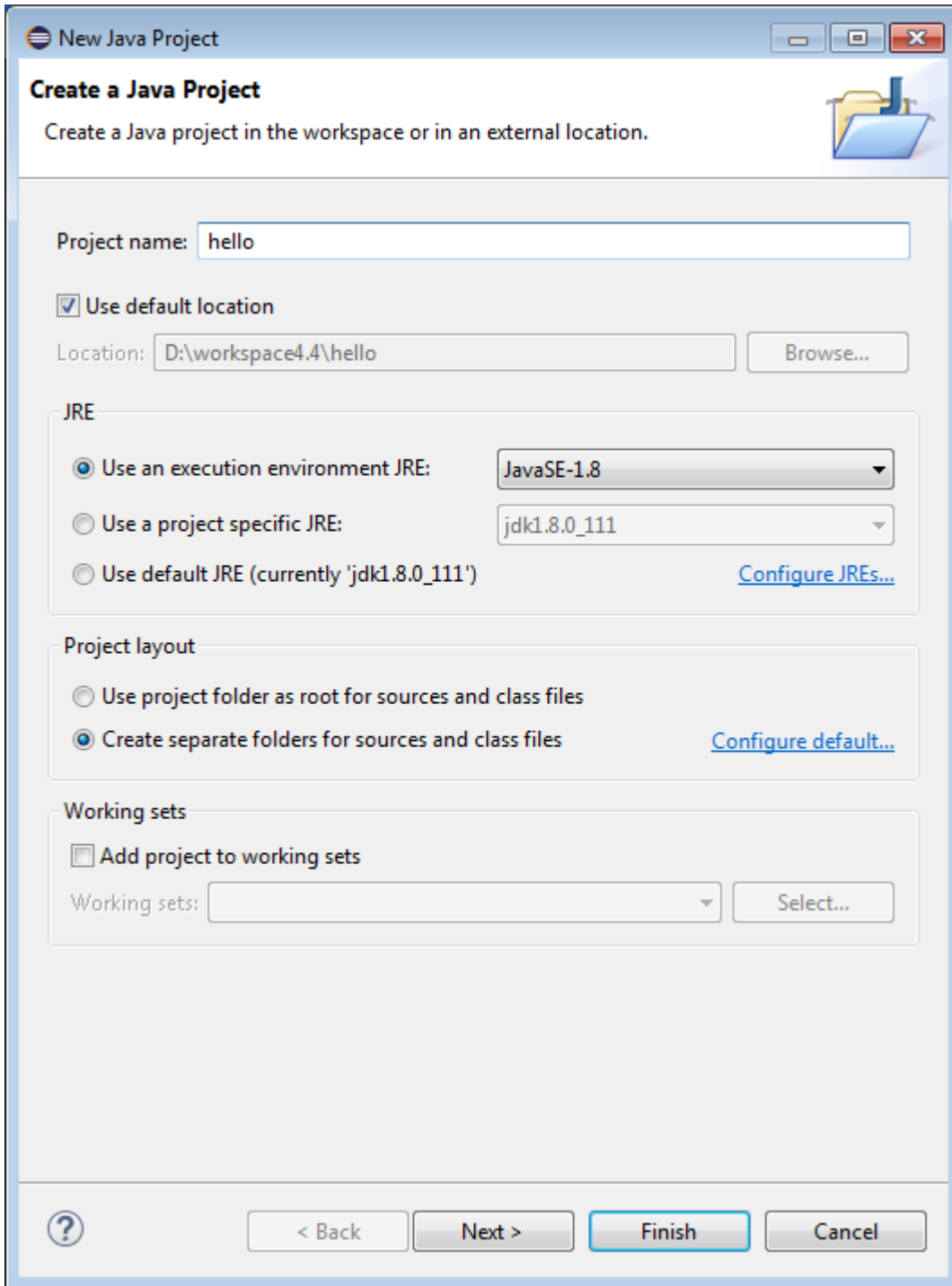
Dans la barre d'outils, ouvrez la perspective Java.



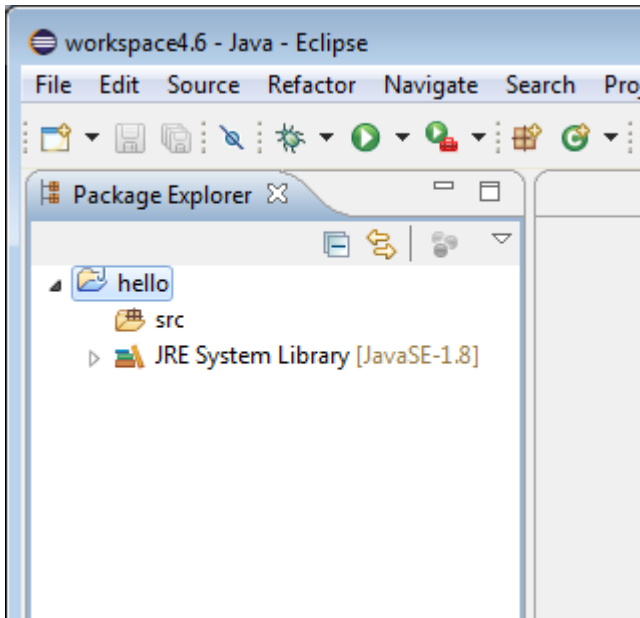
### Créer un nouveau projet Java

Cliquez avec le bouton droit dans l'Explorateur de packages, puis sélectionnez `New -> Java Project` dans le menu `New -> Java Project`

Dans la boîte de dialogue suivante, entrez un nom de projet, puis cliquez sur `Finish`.



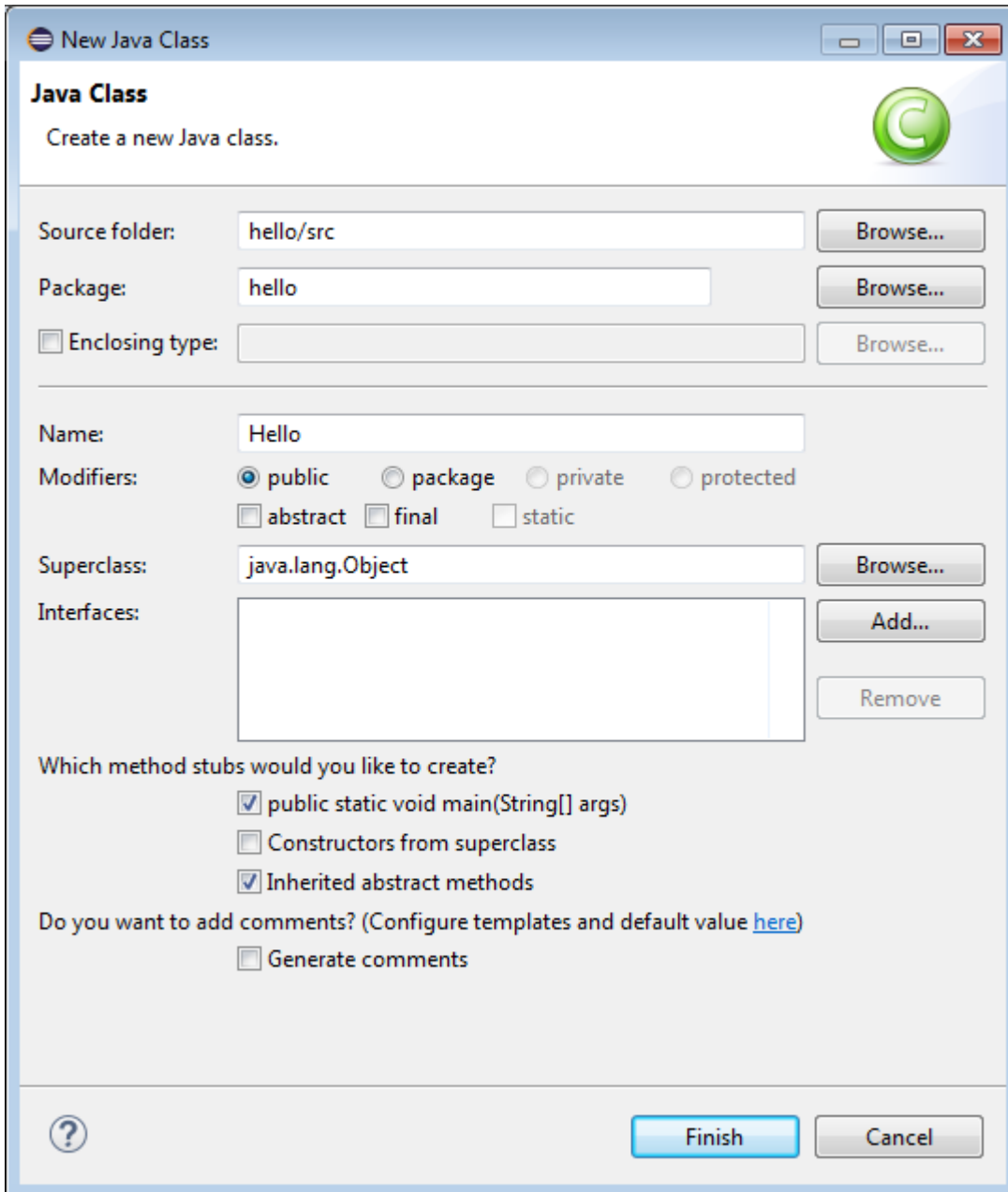
Vous avez maintenant le nouveau projet dans votre espace de travail.



## Créer une nouvelle classe Java

Cliquez avec le bouton droit sur votre projet et dans le menu, sélectionnez `New -> Class`.

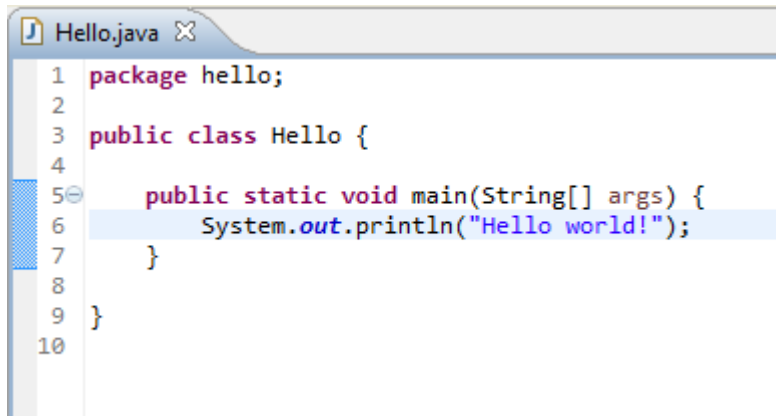
Dans la boîte de dialogue, saisissez le nom de la classe (il doit commencer par une majuscule), sélectionnez également la case à cocher `public static void main(String[] args)`, puis cliquez sur `Finish`.



Vous avez maintenant le premier fichier Java dans votre projet. L'éditeur ouvrira automatiquement ce nouveau fichier.

```
Hello.java
1 package hello;
2
3 public class Hello {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
11
```

Dans la méthode `main`, tapez un code pour imprimer `Hello world!`.

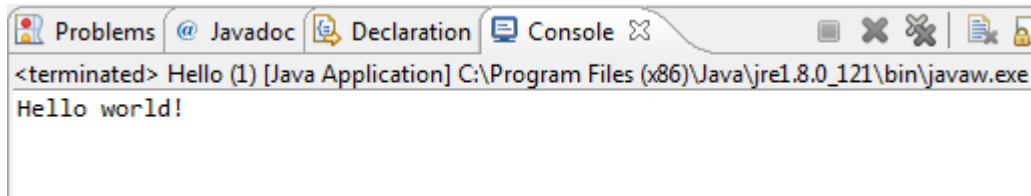


```
1 package hello;
2
3 public class Hello {
4
5     public static void main(String[] args) {
6         System.out.println("Hello world!");
7     }
8
9 }
10
```

## Exécuter votre classe Java

Cliquez avec le bouton droit sur votre classe Java et dans le menu, sélectionnez `Run as -> Java application`.

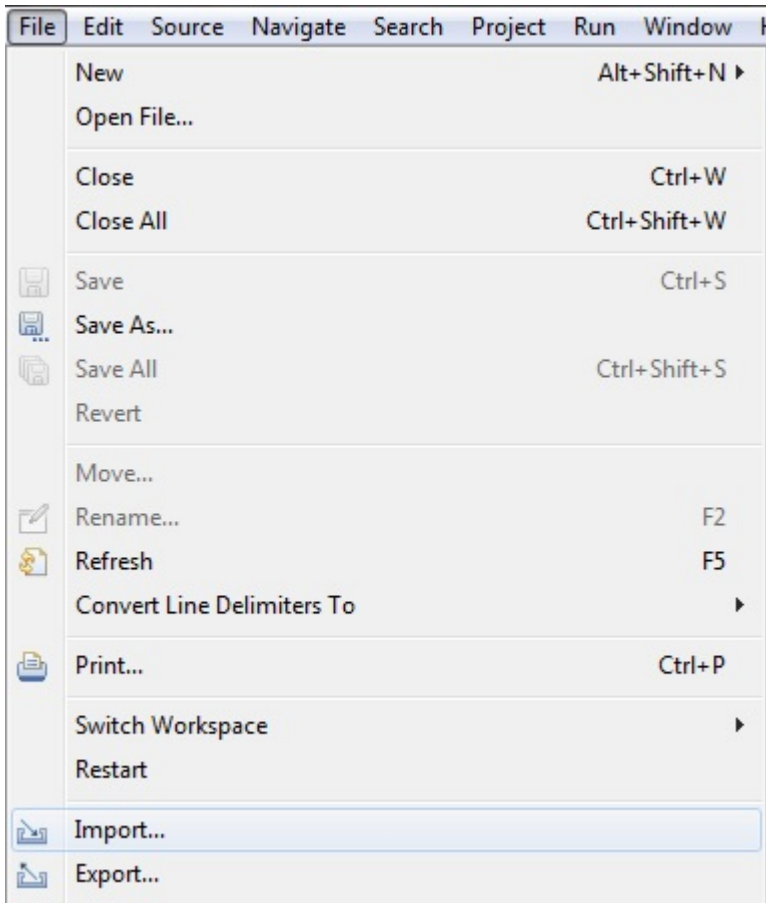
Voilà, vous voyez la sortie de votre programme Java dans la console.



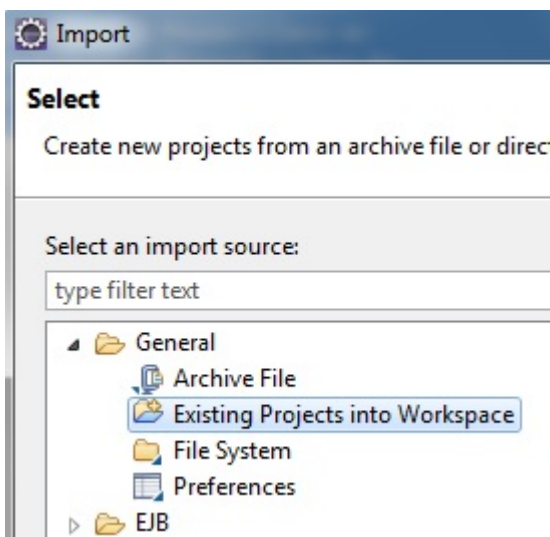
```
<terminated> Hello (1) [Java Application] C:\Program Files (x86)\Java\jre1.8.0_121\bin\javaw.exe
Hello world!
```

## Importation de projets existants

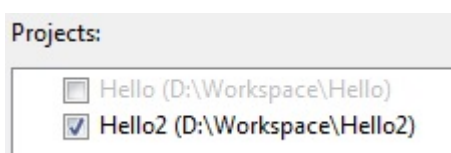
Dans le menu Fichier, choisissez l'option "Importer ...".



Cela ouvre la boîte de dialogue Importer, qui demande le type de projet / fichier que vous souhaitez importer. Pour un projet Java de base, choisissez "Projets existants dans l'espace de travail" dans le dossier "Général".

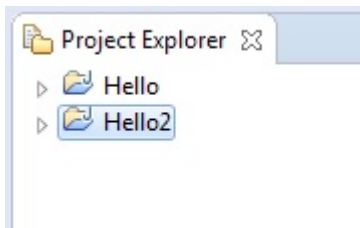


Ensuite, sélectionnez le répertoire où se trouvent les projets en utilisant le bouton "Parcourir". Tous les projets pouvant être importés dans Eclipse apparaîtront dans la section "Projets:". Si le projet a déjà été importé, il sera toujours affiché mais la case à cocher sera désactivée.



Vous pouvez également importer des projets directement à partir d'un fichier compressé en choisissant «Sélectionner le fichier d'archive», puis en cliquant sur le bouton «Parcourir».

Une fois que vous cliquez sur «Terminer», le projet est maintenant visible dans votre explorateur de projet et prêt à être utilisé.



Lire Se lancer dans l'éclipse en ligne: <https://riptutorial.com/fr/eclipse/topic/1143/se-lancer-dans-l-eclipse>

---

# Chapitre 2: Comment Eclipse Remote Debugging fonctionne en coulisse

## Exemples

### Comment fonctionne Eclipse Remote Debugging derrière les scènes

Le débogage Eclipse commence par ce que l'on appelle les agents.

La **JVM**, qui exécute les sources `.class` conformes, possède une fonctionnalité qui permet d'injecter des bibliothèques externes (écrites en Java ou en C++) dans la machine virtuelle Java, à peu près à l'exécution. Ces bibliothèques externes sont appelées agents et peuvent modifier le contenu des fichiers `.class` exécutés. Ces agents ont accès à des fonctionnalités de la machine virtuelle Java qui ne sont pas accessibles à partir d'un code Java régulier en cours d'exécution dans la machine virtuelle Java et ils peuvent être utilisés pour faire des choses intéressantes comme l'injection et modifier le *code source* en cours d'exécution, le *profilage* etc. Des outils tels que **JRebel** utilise cette fonctionnalité pour réaliser leur magie.

Et pour passer un Agent Lib à une JVM, vous le faites via des arguments de démarrage, en utilisant le

```
agentlib:libname[=options] format.
```

Nous transmettons en fait un agent Lib nommé `jdwp` à la JVM exécutant Tomcat. `jdwp` est une implémentation facultative, spécifique à JVM, du *protocole JDWP (Java Debug Wire Protocol)* utilisé pour définir la communication entre un débogueur et une machine virtuelle Java en cours d'exécution. Son implémentation, si elle est présente, est fournie en tant que bibliothèque native de la JVM en tant que `jdwp.so` ou `jdwp.dll`

### Alors qu'est-ce que ça fait?

En termes simples, l'agent `jdwp` que nous transmettons sert essentiellement de lien entre l'instance JVM exécutant une application et un débogueur (qui peut être distant ou local). Comme il s'agit d'une bibliothèque d'agents, il est possible d'intercepter le code en cours d'exécution, de créer un pont entre la machine virtuelle Java et un débogueur, et d'appliquer les fonctionnalités d'un débogueur à la machine virtuelle Java. Dans l'architecture JVM, la fonctionnalité de débogage ne se trouve pas dans la JVM elle-même, mais elle est abstraite dans des outils externes (appelés à juste titre les débogueurs). Ces outils peuvent soit résider sur la machine locale à partir de la machine externe. C'est cette architecture modulaire découplée qui nous permet d'avoir une JVM exécutée sur une machine distante et utilisant le JDWP, de faire communiquer avec un débogueur distant.

En bref, voici comment fonctionne le débogueur Eclipse.

Lire Comment Eclipse Remote Debugging fonctionne en coulisse en ligne:

<https://riptutorial.com/fr/eclipse/topic/6247/comment-eclipse-remote-debugging-fonctionne-en->



coulisse

# Chapitre 3: Configuration d'Eclipse

## Exemples

### Augmentation de la mémoire de tas maximale pour Eclipse

Pour augmenter la quantité maximale de mémoire de segment utilisée Eclipse, modifiez le fichier `eclipse.ini` situé dans le répertoire d'installation d'Eclipse.

Ce fichier spécifie les options pour le démarrage d'Eclipse, telles que la JVM à utiliser et les options pour la JVM. Plus précisément, vous devez modifier la valeur de l'option JVM `-Xmx` (ou en créer une si elle n'existe pas).

Vous trouverez ci-dessous un exemple de configuration qui définit une mémoire de pile maximale de 1 Go (1024 m). La ligne correspondante est `-Xmx1024m` - cela remplacerait la ligne `-Xmx*` existante dans votre configuration:

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.200.v20160318-1642.jar
--launcher.library
C:/Users/user1/.p2/pool/plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.400.v20160518-1444
-product
org.eclipse.epp.package.java.product
--launcher.defaultAction
openFile
-showsplash
org.eclipse.platform
--launcher.appendVmargs
-vmargs
-Xms256m
-Xmx1024m
```

### Spécifier la JVM

Un problème courant rencontré par les utilisateurs d'Eclipse est lié à la machine virtuelle Java par défaut du système.

Une situation typique est un Windows 64 bits avec les versions 32 et 64 bits de Java installées et un Eclipse 32 bits. Si la version 64 bits de Java est la valeur par défaut du système, une boîte de dialogue d'erreur s'affiche lors du lancement d'Eclipse.

Spécifier explicitement la JVM dans `eclipse.ini` résoudra ce problème. L'entrée `-vm` doit être ajoutée directement au-dessus de la section `-vmargs`.

L'exemple ci-dessous montre comment utiliser une JVM 32 bits sur un Windows 64 bits:

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.200.v20160318-1642.jar
```

```
...  
-vm  
C:/Program Files (x86)/Java/jdk1.7.0_71/bin/javaw.exe  
-vmargs  
-Xms256m  
-Xmx1024m
```

## Comment configurer la taille de police des vues dans Eclipse sous Linux

Eclipse ne vous donne pas la possibilité de modifier la taille de police des vues comme "Explorateur de projet" ou "Serveurs", ce qui est moche sous Linux car Eclipse utilise la taille de police par défaut (bureau). Mais vous pouvez éditer des fichiers de configuration spécifiques pour obtenir les tailles de police appropriées.

Pour corriger cette taille de police ennuyeuse, allez à

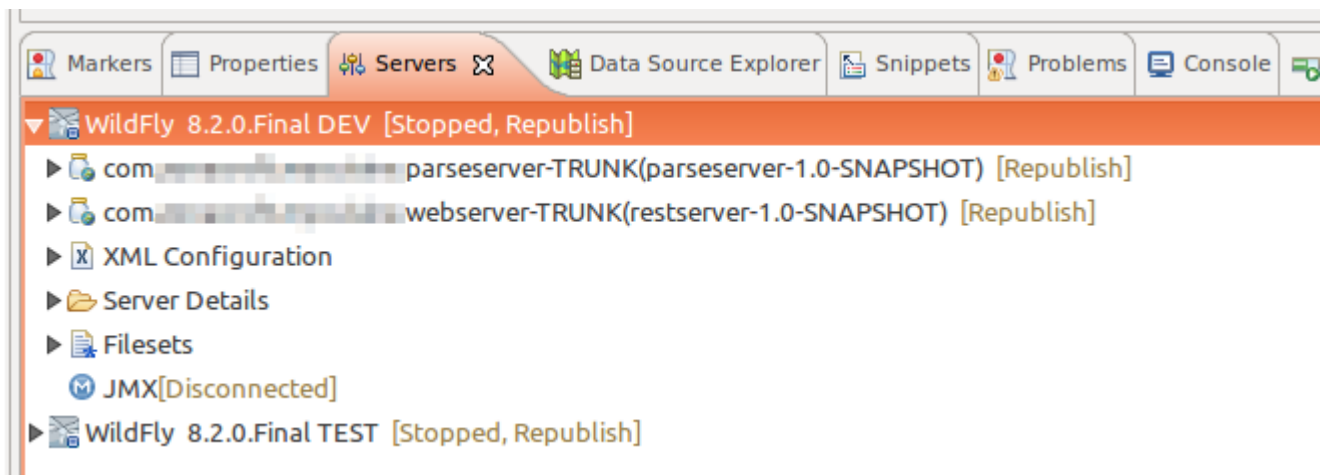
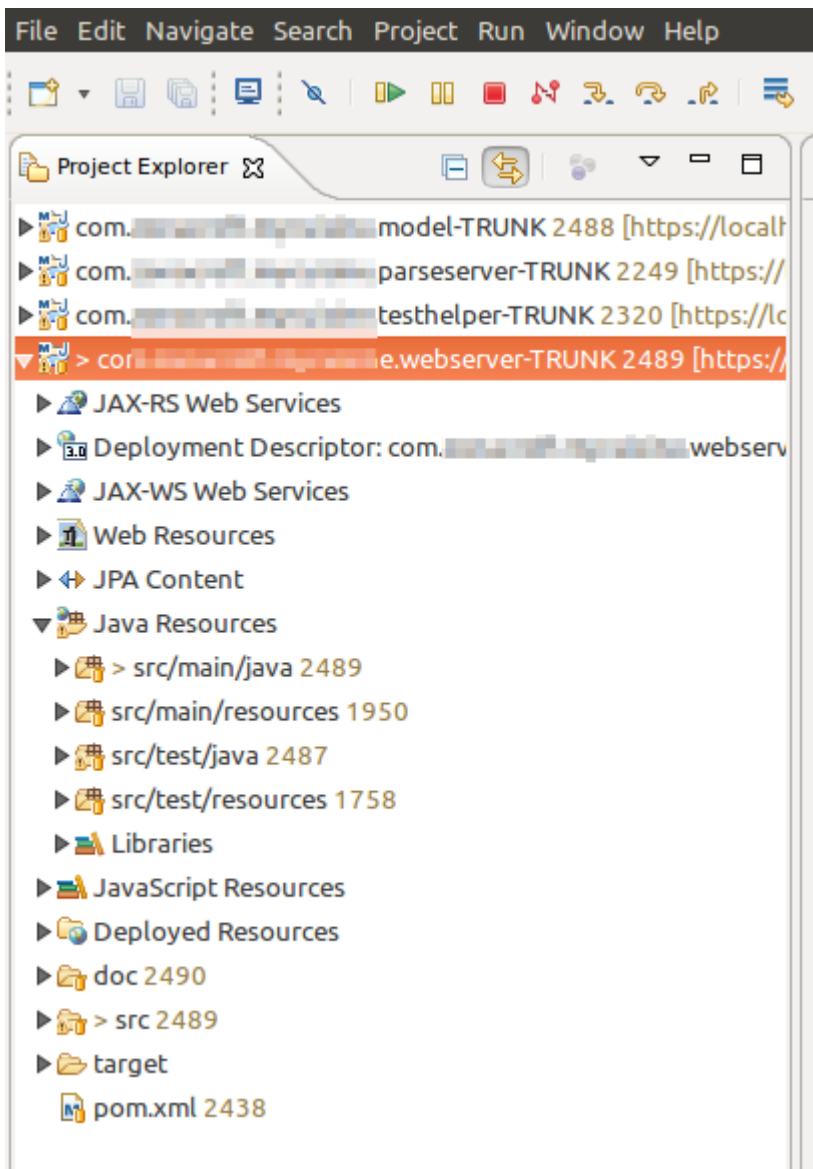
`/[YOUR_INST_DIR]/eclipse/plugins/org.eclipse.ui.themes_[LATEST_INSTALLATION]/css` et ajoutez ce contenu ...

```
.MPart Tree{  
    font-family: Sans;  
    font-size: 8px;  
}
```

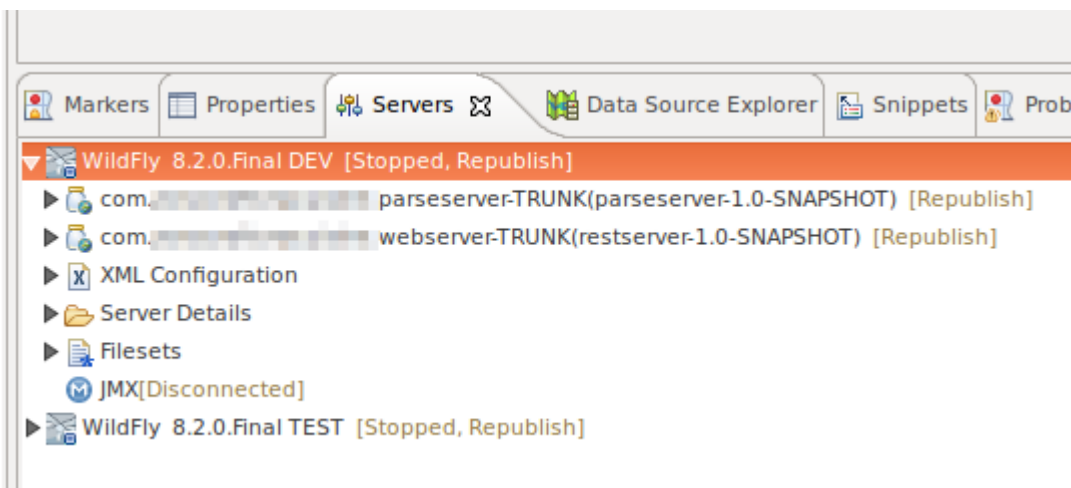
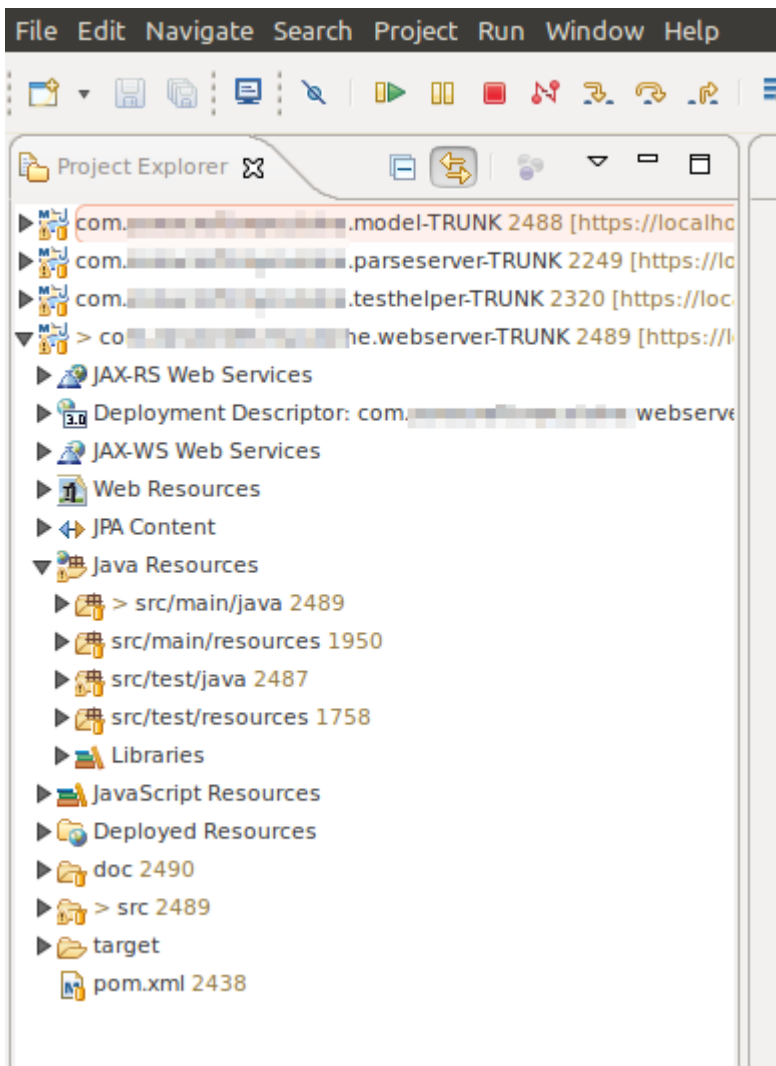
au bas des fichiers suivants:

```
e4_classic_winxp.css  
e4_classic_win7.css
```

### AVANT DE CHANGER



## APRES CHANGEMENT



Lire Configuration d'Eclipse en ligne: <https://riptutorial.com/fr/eclipse/topic/2112/configuration-d-eclipse>

# Chapitre 4: Configurer Eclipse pour C ++

## Exemples

### Linux + CMake (générateur "Makefiles Unix") + Qt (optionnel)

Vous devriez avoir un projet CMake simple **myproject** , et nous allons créer un espace de travail Eclipse en dehors de celui-ci:

```
myproject/  
  .git/  
  CMakeLists.txt  
  src/  
    main.cpp  
workspace/  
  myproject/  
    Release/  
    Debug/
```

## Qt (optionnel)

- Procurez-vous le dernier CDT Eclipse, puis installez le package Qt via "Aide -> Installer un nouveau logiciel".

## Espace de travail

- Créez un répertoire "espace de travail" vide à côté de votre répertoire source de projet CMake.
- Lancez Eclipse et basculez vers ce répertoire "workspace".
- Créez un projet C ++ (pour Qt avec Eclipse plus ancien que Neon: créez "Qt Makefile Project" puis supprimez-y les fichiers \* .pro, makefile et main.cpp)

## Attacher des sources au projet

- Accédez à Propriétés du projet -> Chemins et symboles -> Emplacement source -> Dossier de liaison.
- Cochez "Avancé" et liez le dossier source du projet CMake comme ceci:  
../.. /myproject/src/ . Cela fonctionne parce que l'espace de travail est juste en dehors du répertoire du projet CMake.

## Générateur de CMake

- Créez le dossier `Release` dans le projet.
- Allez dans la vue "Make Target" ( `Ctrl + 3` puis tapez "Make Target" si c'est difficile à trouver). La vue "Make Target" ressemble à la vue du projet.

- Faites un clic droit sur le dossier "Release" puis cliquez sur "New ...".
  - Désélectionnez "Identique au nom de la cible".
  - Désélectionnez "Utiliser les paramètres du générateur".
  - Tapez "Release" dans le champ "Nom de la cible".
  - Laissez "Make target" vide.
  - Réglez "Build command" sur quelque chose comme `cmake ../../../../myproject/`.
  - Cliquez sur OK.
- Double-cliquez sur cette "Release" pour créer la cible qui vient d'être créée dans le dossier Release. Cela va lancer la génération de cmake.

## Construire

- Accédez aux propriétés du projet et créez une configuration "Release".
- Rendre la configuration "Release" active.
- Pour la configuration "Release", décochez "Générer automatiquement les Makefiles".
- Définissez le répertoire de construction sur "Release".
- Activer la construction parallèle

Maintenant, vous pouvez construire le projet à partir d'Eclipse avec un `Ctrl + b` "Build" habituel.

## Ré-exécuter CMake (pour générer à nouveau les makefiles)

- Tout supprimer du répertoire "Release".
- Allez dans la vue "Make Target".
- Double-cliquez sur la cible "Release".

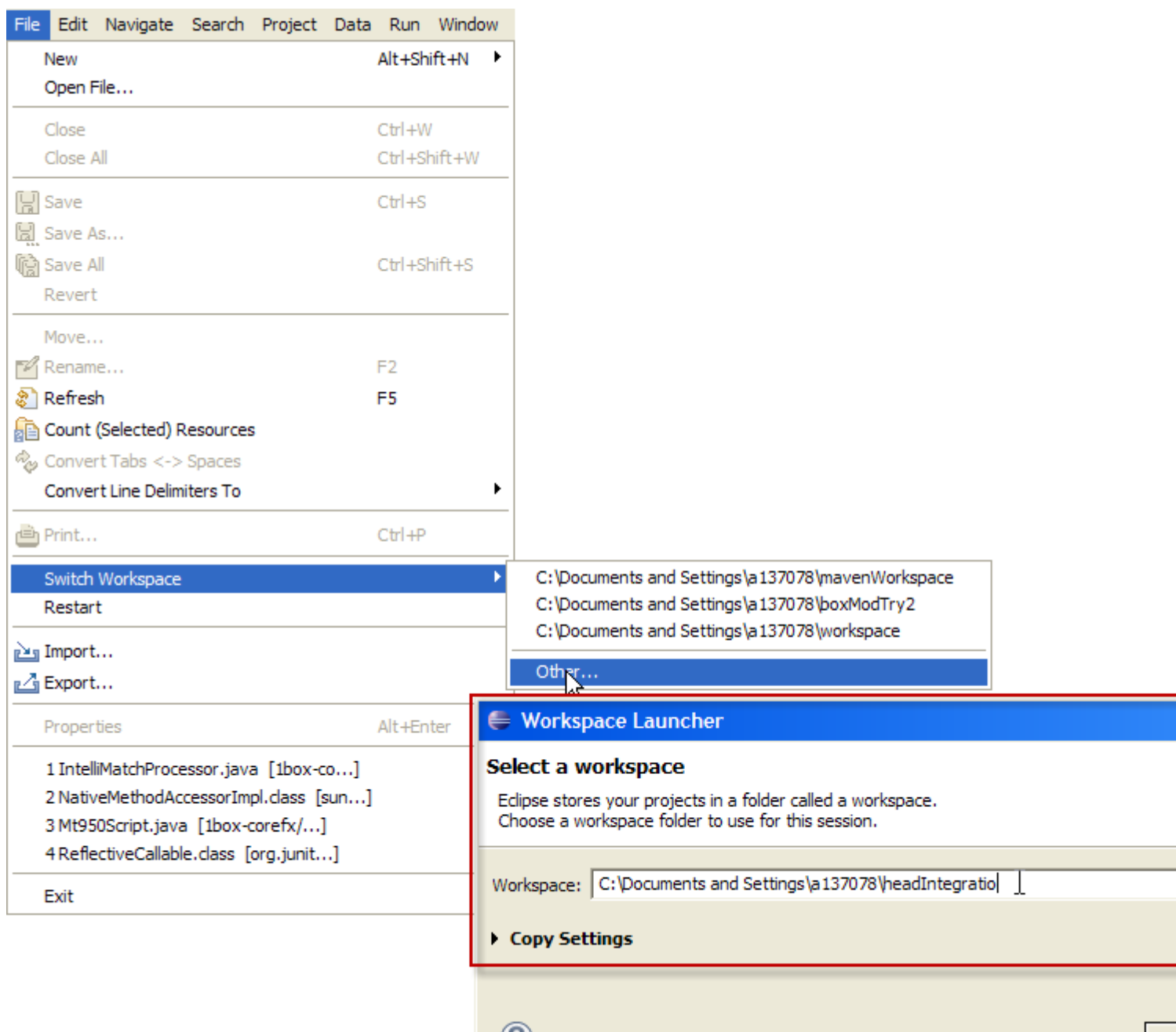
Lire Configurer Eclipse pour C ++ en ligne: <https://riptutorial.com/fr/eclipse/topic/7028/configurer-eclipse-pour-c-plusplus>

# Chapitre 5: Créer un nouvel espace de travail dans Eclipse

## Exemples

### Comment créer un espace de travail

Allez dans Fichier -> Changer d'espace de travail -> Autre ... et tapez votre nouveau nom d'espace de travail.



Lire Créer un nouvel espace de travail dans Eclipse en ligne:

<https://riptutorial.com/fr/eclipse/topic/6345/creer-un-nouvel-espace-de-travail-dans-eclipse>



---

# Chapitre 6: Débogage à distance dans Eclipse

## Exemples

### Configurer le débogage à distance Eclipse pour une application

Voici les étapes à suivre pour démarrer un débogueur distant Eclipse. Ceci est utile lorsque l'application n'est pas démarrée à partir d'une instance de serveur dans Eclipse. Cette fonctionnalité est très puissante et peut également aider à déboguer du code qui réside dans l'environnement de test ou de production. Regardons les paramètres:

#### Paramètres Eclipse:

1. Cliquez sur le bouton Exécuter
2. Sélectionnez les configurations de débogage
3. Sélectionnez "Application Java distante"
4. Nouvelle configuration
  - a) Nom: GatewayPortalProject
  - b) Projet: GatewayPortal-portlet
  - c) Type de connexion: Socket Attach
  - d) Propriétés de connexion:
    - i) localhost ii) 8787

#### Pour JBoss:

Modifiez le `/path/to/Jboss/jboss-eap-6.1/bin/standalone.conf` dans votre vm comme suit:  
Décommentez la ligne suivante en supprimant le #:

```
JAVA_OPTS="$JAVA_OPTS -agentlib:jdwp=transport=dt_socket,address=8787,server=y,suspend=n"
```

#### Pour Tomcat:

Dans le fichier `catalina.bat` :

Étape 1:

```
CATALINA_OPTS="-Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n"
```

Étape 2:

```
JPDA_OPTS="-agentlib:jdwp=transport=dt_socket,address=8000,server=y,suspend=n"
```

Étape 3: Exécutez Tomcat à partir de l'invite de commande comme ci-dessous:

```
catalina.sh jpda start
```

Ensuite, vous devez définir des points d'arrêt dans les classes Java que vous souhaitez déboguer.

Lire [Débogage à distance dans Eclipse en ligne](#):

<https://riptutorial.com/fr/eclipse/topic/3502/debogage-a-distance-dans-eclipse>

# Chapitre 7: Débogage des programmes Java dans Eclipse

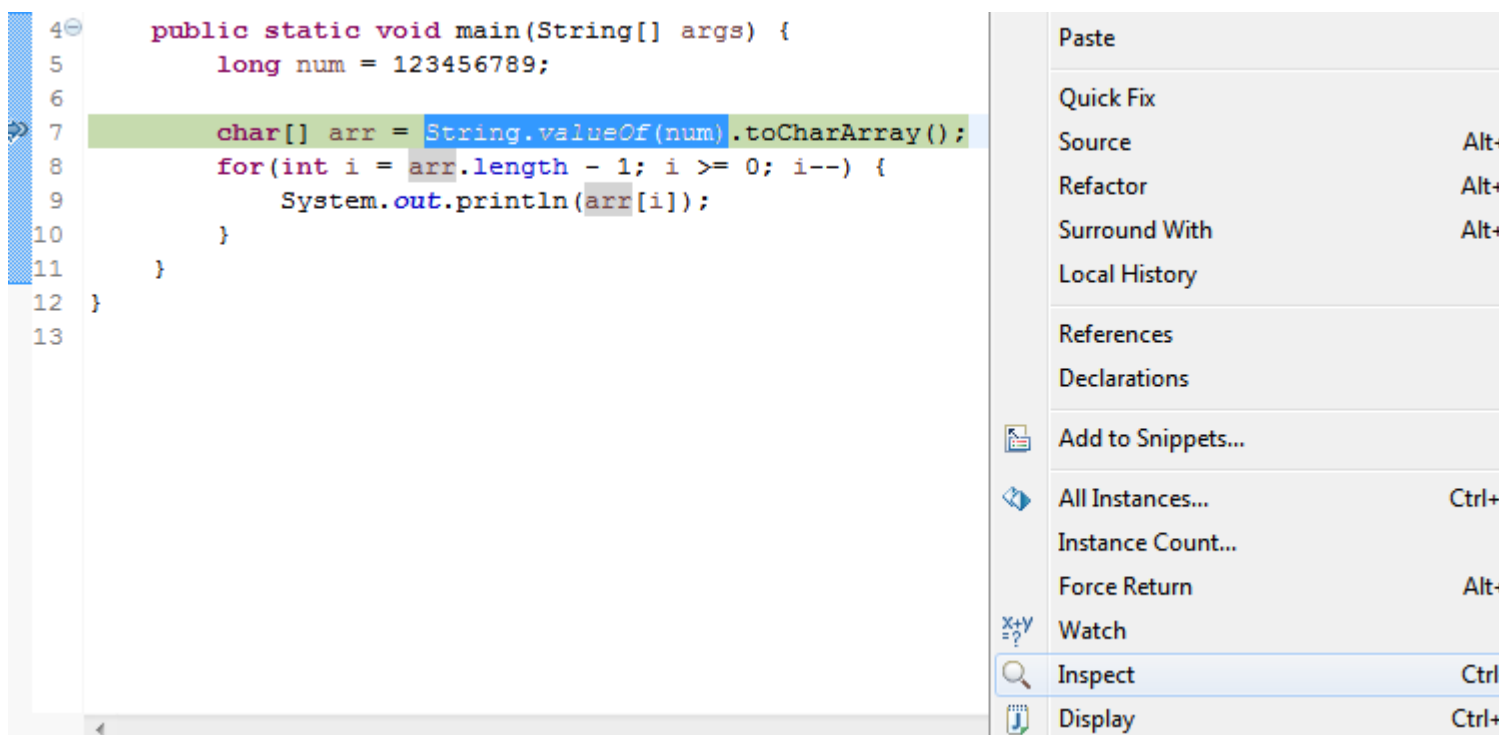
## Exemples

### Évaluation d'expressions dans une session de débogage

Il y en a plusieurs pour évaluer une certaine expression lors du débogage d'une application Java.

#### 1. Inspecter manuellement une expression

Lorsque l'exécution du programme est suspendue à une certaine ligne (soit en raison d'un point d'arrêt ou pas à pas manuellement via le débogueur), vous pouvez manuellement évaluer une expression en sélectionnant l'expression dans le code, puis un clic droit et sélectionner **Inspect** comme indiqué dans le ci - dessous capture d'écran. Sinon, faites `Ctrl + Shift + I` après avoir sélectionné l'expression.



#### 2. Regarder une expression dans la vue Expressions

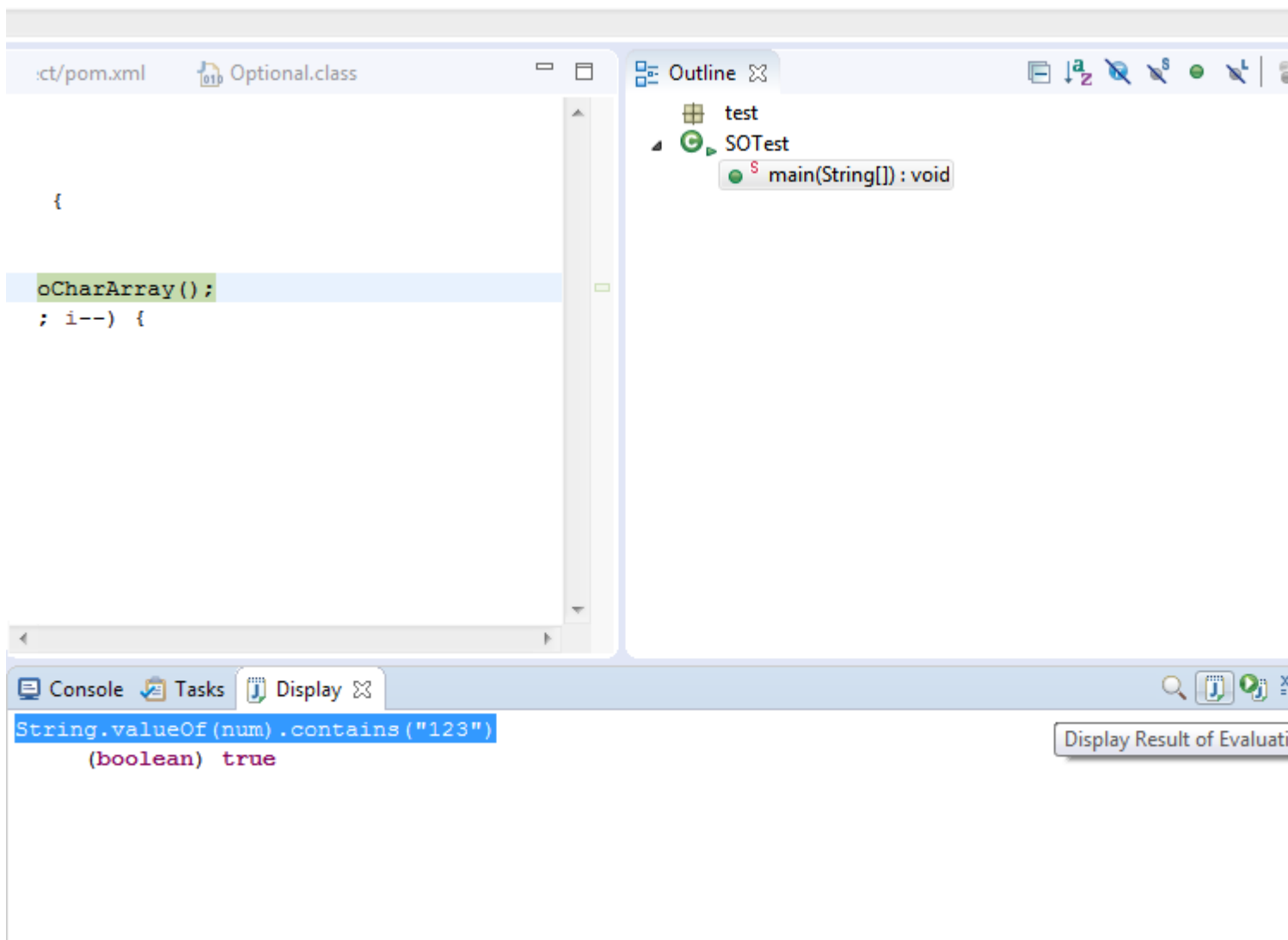
Si vous souhaitez évaluer en permanence une expression, par exemple parce qu'elle se trouve dans une boucle, vous pouvez la regarder dans la vue Expressions. De cette façon, sa valeur sera affichée chaque fois que le programme est suspendu à chaque itération de la boucle. Pour ce faire, sélectionnez l'expression souhaitée, puis cliquez avec le bouton droit et sélectionnez **Regarder**. Cela ouvrira la vue **Expressions** et affichera la valeur de l'expression (voir l'image ci-dessous). Vous pouvez également écrire manuellement l'expression dans la vue.

Name	Value
$x+y$ "String.valueOf(num)"	123456789
<a href="#">+ Add new expression</a>	

### 3. Utilisation de la vue Display pour évaluer et exécuter des instructions

La vue d' **affichage** vous permet d'écrire vos propres expressions, instructions ou tout code en général qui serait évalué ou exécuté en contexte avec le code du programme suspendu. Cela peut être utile si vous souhaitez évaluer des expressions complexes sans modifier votre original et redémarrer le débogage.

Pour ouvrir la vue Affichage, sélectionnez **Fenêtre> Afficher la vue> Affichage** . Ensuite, écrivez votre expression ou vos instructions dans la vue et sélectionnez l'une des options de la barre d'outils de la vue, par exemple pour exécuter les instructions écrites, ou affichez le résultat de leur évaluation dans la vue, comme illustré ci-dessous. Le code écrit dans la vue d'affichage peut également être inspecté ou exécuté en le sélectionnant, puis en cliquant avec le bouton droit de la souris et en sélectionnant l'action souhaitée.



## Débogage à distance d'une application Java

Afin de déboguer une application Java distante, il doit être lancé avec des arguments supplémentaires pour demander à la JVM de l'exécuter en mode débogage. Cela se fait comme suit:

```
java -Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=8000 -jar sampleApp.jar
```

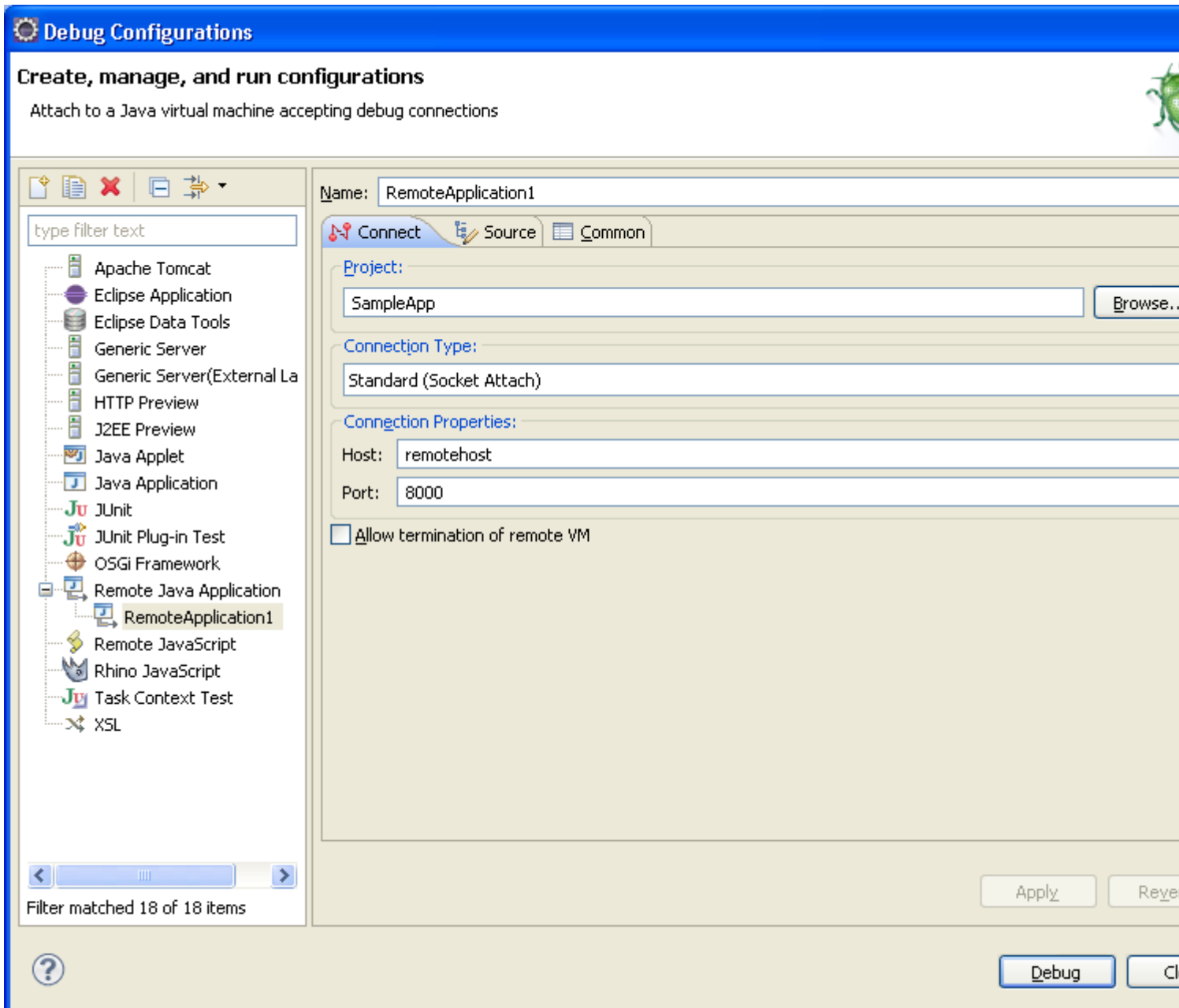
La commande ci-dessus indique à la JVM de démarrer l'application sampleApp.jar tout en ayant un socket serveur à l'écoute sur le port 8000 pour qu'un débogueur s'y connecte. L'argument suspend indique si vous souhaitez que la machine virtuelle Java attende qu'un débogueur soit connecté au numéro de port avant que l'application ne s'exécute effectivement.

Après avoir lancé l'application distante avec les paramètres ci-dessus, l'étape suivante consiste à connecter votre débogueur Eclipse à l'application distante. Cela se fait comme suit:

1. Allez à **Exécuter > Configurations de débogage ...**
2. Dans la boîte de dialogue **Configurations de débogage** (voir la figure ci-dessous), sélectionnez la section **Application Java distante** et cliquez sur **Nouvelle** action de

## configuration du lancement .

3. Donnez un nom à votre configuration de débogage, puis sélectionnez le projet contenant le code de votre application distante. Définissez le nom d'hôte ou l'adresse IP de la machine distante et le port auquel le débogueur doit se connecter.
4. Cliquez sur **Déboguer** . Le débogueur Eclipse va maintenant tenter de se connecter à l'application distante et suspendre aux points d'arrêt définis dans le code source.



Lire Débogage des programmes Java dans Eclipse en ligne:

<https://riptutorial.com/fr/eclipse/topic/4548/debogage-des-programmes-java-dans-eclipse>

---

# Chapitre 8: Procédure de déploiement Tomcat

## Exemples

### Procédure quand rien d'autre n'aide

Une fois de temps en temps, le concecutent se déploie sur tomcat interne et commence à donner une erreur constante, sans cause claire (début de l'écouteur ou exception `ClassNotFoundException`). Lorsque rien ne semble le guérir, cette procédure sauve le monde:

- 1 supprimer le dossier Serveurs
- 2 redémarrer Eclipse
- 3 créer un nouveau serveur, ajouter un projet et démarrer

Fonctionne comme un charme et n'est pas si long.

Si, pour une raison quelconque, cela échoue, ma longue procédure initiale (où l'autre est un raccourci qui devrait faire la même chose) est ici:

- 1 serveur d'arrêt
- 2 projet -> propre
- 3 projet build (j'ai eu la construction automatique désactivée)
- 4 supprimer le serveur
- 5 supprimer le dossier Serveurs
- 6 redémarrer Eclipse
- 7 créer un nouveau serveur, ajouter un projet et démarrer.

Avec cette opération en sept étapes, les problèmes de déploiement ne sortent jamais de votre code et de votre contrôle.

### Remarque:

*Vous n'avez pas besoin d'autre chose que le rafraîchissement de la page si tout se passe bien. Cette procédure est effectuée une fois par message d'erreur pour être sûr de vous débarrasser de l'erreur incertaine, si votre code semble correct et que vous n'avez rien fait pour recevoir l'erreur. L'erreur contient le mot `ClassNotFoundException` ou `ListenerStart`, en fonction de l'environnement utilisé. Notez également que cela ne résout pas les exceptions `ClassNotFoundExceptions` causées par des bibliothèques manquantes dans un projet.*

Lire Procédure de déploiement Tomcat en ligne:

<https://riptutorial.com/fr/eclipse/topic/6092/procedure-de-deploiement-tomcat>



---

# Chapitre 9: Raccourcis Eclipse

## Introduction

Eclipse a de nombreux raccourcis pour vous simplifier la vie.

## Exemples

### Code commentaire / décomment

Pour commenter ou décommenter le code, sélectionnez les lignes et utilisez **Ctrl + Maj + C** ou **Ctrl + Maj + /**

### Ouvrir le dialogue de recherche

Pour accéder à la boîte de dialogue Ouvrir une ressource, utilisez **Ctrl + Maj + R**. De là, vous pouvez commencer à taper un nom de ressource et il trouvera toutes les correspondances dans l'espace de travail, ce qui facilite la recherche d'un fichier lorsque vous ne le savez pas exactement.

### Pour obtenir une impression

Si vous voulez `System.out.println ();` mais ne voulez pas taper le tout, vous pouvez simplement taper **syso** et **appuyer** sur **Ctrl + Espace** . Il va taper le reste et placer le curseur entre les parenthèses.

### Générer des Getters et des Setters

Eclipse peut générer des getters et des setters de base pour vous. Cliquez avec le bouton droit de la souris sur le fichier de votre classe et allez à **Source - Générer des objets Getters et Setters** (ALT + SHIFT + S). Cela ouvrira une boîte de dialogue dans laquelle vous pourrez choisir les champs pour lesquels vous souhaitez générer des getters et des setters.

### Texte mis en évidence par le refactor

Renommer une variable ou une classe est généralement une tâche fastidieuse, en recherchant tous les emplacements où il est utilisé. Cela peut être considérablement accéléré en mettant en évidence le mot, en appuyant sur **Alt + Maj + R** , puis en tapant le nouveau mot. Eclipse renommera automatiquement le mot dans chaque fichier où il est appelé.

### Format xml

Lorsque vous ajoutez des entrées à un fichier XML ou à une copie provenant d'autres sources, les entrées et les espaces sont souvent inégaux.

Lorsque vous appuyez sur **Ctrl + Maj + F** , vous alignez facilement tout le document et supprimez également les onglets supplémentaires. Ainsi, le texte est formaté et devient éventuellement lisible.

Lire Raccourcis Eclipse en ligne: <https://riptutorial.com/fr/eclipse/topic/9387/raccourcis-eclipse>

# Crédits

S. No	Chapitres	Contributeurs
1	Se lancer dans l'éclipse	<a href="#">Aaron Vigal</a> , <a href="#">Ala Eddine JEBALI</a> , <a href="#">Aleksandr M</a> , <a href="#">Ani Menon</a> , <a href="#">Community</a> , <a href="#">Latsuj</a> , <a href="#">manouti</a> , <a href="#">Ray</a> , <a href="#">Thomas Fritsch</a>
2	Comment Eclipse Remote Debugging fonctionne en coulisse	<a href="#">Pritam Banerjee</a>
3	Configuration d'Eclipse	<a href="#">Bevor</a> , <a href="#">manouti</a> , <a href="#">mecscsco</a> , <a href="#">romeara</a>
4	Configurer Eclipse pour C ++	<a href="#">Velkan</a>
5	Créer un nouvel espace de travail dans Eclipse	<a href="#">rajah9</a>
6	Débogage à distance dans Eclipse	<a href="#">Pritam Banerjee</a>
7	Débogage des programmes Java dans Eclipse	<a href="#">manouti</a>
8	Procédure de déploiement Tomcat	<a href="#">mico</a>
9	Raccourcis Eclipse	<a href="#">Latsuj</a> , <a href="#">Ray</a> , <a href="#">Srishti Sinha</a> , <a href="#">user7491506</a> , <a href="#">Yurii COjocari</a>