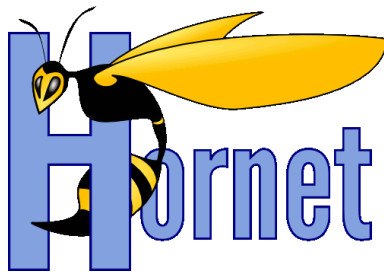




[HORNET]

Guide de paramétrage



Guide de paramétrage de l'environnement de développement

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique disponible en ligne <http://creativecommons.org/licenses/by-nc-sa/2.0/fr/> ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA

SUIVI DES MODIFICATIONS

Version	Auteur	Description	Vérification	Date
1.0	O. Coatanlem	Initialisation du document	P.Y. Baloché S. Heurtematte	02/05/2013
1.1	S. Percier	Ajout forme de fin de ligne (Unix)	S.Heurtematte	21/05/2013
1.2	S.Heurtematte	Mise à jour eclipse kepler	S. Percier	24/10/2013
1.3	O. Rousseil	Mises à jour de version		
1.4	O. Rousseil	Mises à jour de version		

DOCUMENTS DE REFERENCE

Document

HORNET_GUI_Guide du développeur Hornet
HORNET_NOR_Normes d'encodage
HORNET_GUI_GuideCréationProjet

SOMMAIRE

SUIVI DES MODIFICATIONS	2
DOCUMENTS DE REFERENCE	2
SOMMAIRE	3
TABLEAUX	4
FIGURES	4
1 INTRODUCTION	5
1.1 OBJECTIF DU DOCUMENT.....	5
1.2 COMPOSITION DE L'ENVIRONNEMENT HORNET.....	5
2 CONFIGURATION DE L'ENVIRONNEMENT D'EXECUTION	6
2.1 JAVA DEVELOPMENT KIT 1.6.....	6
2.1.1 <i>Installation</i>	6
2.2 APACHE TOMCAT 6.0.....	6
2.2.1 <i>Installation</i>	6
2.3 APACHE HTTP SERVER 2.2.....	6
2.3.1 <i>Installation</i>	6
2.3.2 <i>Configuration</i>	8
3 IDE ECLIPSE	10
3.1 INSTALLATION.....	10
3.2 CONFIGURATION.....	10
3.2.1 <i>Lancement</i>	10
3.2.2 <i>Création du workspace</i>	10
3.2.3 <i>Paramétrage du JDK</i>	11
3.2.4 <i>Environnement d'exécution Tomcat</i>	13
3.2.5 <i>Encoding (et format de fin de ligne)</i>	14
3.2.6 <i>Apache Ant</i>	16
3.3 PARAMETRAGE HORNET.....	17
3.3.1 <i>Présentation</i>	17
3.3.2 <i>Gestionnaire d'installation</i>	17
3.3.3 <i>PMD</i>	17
3.3.4 <i>FindBugs</i>	19
3.3.5 <i>Checkstyle</i>	20
3.3.1 <i>Formatage et modèles</i>	25
3.4 PLUGINS COMPLEMENTAIRES.....	27
3.4.1 <i>Présentation</i>	27
3.4.2 <i>Subclipse</i>	28
3.5 AJOUT D'UN SERVEUR TOMCAT POUR LE DEVELOPPEMENT.....	29
4 REPOSITORIES IVY	33
4.1 PRESENTATION D'IVY.....	33
4.2 INSTALLATION ET CONFIGURATION.....	33
4.2.1 <i>En local</i>	33
4.2.2 <i>Depuis un répertoire réseau</i>	33
4.2.1 <i>Depuis un gestionnaire d'artefacts</i>	33
5 COMPOSANTS COMPLEMENTAIRES	35
5.1 INSTALLATION DE GRAPHVIZ.....	35
5.2 VERIFICATION DE GRAPHVIZ.....	35

TABLEAUX

Aucune entrée de table d'illustration n'a été trouvée.

FIGURES

Figure 1 : Configuration lors de l'installation	7
Figure 2 : Choix du type d'installation.....	7
Figure 3 : Page de test du serveur HTTP	8
Figure 4 : Sélection du workspace.....	11
Figure 5 : Définition d'un nouveau JRE.....	12
Figure 6 : Choix du JDK.....	13
Figure 7 : Déclaration du serveur Tomcat	14
Figure 8 : Fenêtre de préférences générales d'Eclipse.....	15
Figure 9 : Fenêtre de préférences d'un fichier dans Eclipse	16
Figure 10 : Gestionnaire d'installation de plugins Eclipse.....	17
Figure 11 : Suppression des règles PMD.....	18
Figure 12 : Plugin Checkstyle : Configuration	21
Figure 13 : Plugin Checkstyle : Référencement de la configuration	22
Figure 14 : Plugin Checkstyle : Etape finale de configuration	23
Figure 15 : Propriété d'un projet : Vérification Checkstyle	24
Figure 16 : Formatage Java : Nouveau profile.....	25
Figure 17 : Modèle Java : Nouveau configuration.....	26
Figure 18 : Formatage Java : Nouveau profile.....	27
Figure 19 : Configuration Subclipse.....	28
Figure 20 : Ajout de la vue « Svn Repositories »	28
Figure 21 : Sélection du nouveau dépôt.....	29
Figure 22 : Récupération locale du nouveau dépôt.....	29
Figure 23 : Afficher la vue "Servers"	30
Figure 24 : Création d'un nouveau serveur de test.....	30
Figure 25 : Paramétrage d'un serveur de test	31
Figure 26 : Choix des applications à ajouter.....	32
Figure 27 : Serveur de test prêt.....	32
Figure 28 : Installation de Graphviz : Emplacement	35

1 Introduction

1.1 Objectif du document

Ce document a pour but de décrire les opérations à effectuer pour installer et configurer un environnement de développement pour Hornet version 3.6 et supérieur.

Ce document décrit les opérations sur système d'exploitation Windows uniquement, il est cependant tout à fait adaptable sur d'autres environnements (Linux notamment).

1.2 Composition de l'environnement Hornet

L'environnement d'exécution est composé des éléments suivants :

- Java 1.6.0
- Tomcat 6.0.35
- Apache 2.2

La gestion des dépendances est assurée par Ivy 2.3.0 et la construction des livrables (war, jar ...) est effectuée grâce à des scripts Ant (pour la version 1.7.1).

L'environnement de développement intégré (IDE) retenu est Eclipse Kepler SR1.

L'application Graphviz (version 2.30.1) est également utilisée pour la phase de génération des documentations Java, particulièrement pour l'intégration de modèles UML.

2 Configuration de l'environnement d'exécution

2.1 Java Development Kit 1.6

2.1.1 Installation

La version de Java supportée par Hornet est la 1.6.0

- Récupérer le binaire correspondant au système d'exploitation de développement depuis l'url <http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html> (ex : jdk-6u43-windows-i586.exe pour Windows).
- Exécuter l'installation en gardant les options par défaut.

2.2 Apache Tomcat 6.0

2.2.1 Installation

La version de Tomcat supportée par Hornet est la 6.0

- Depuis l'url <http://tomcat.apache.org/download-60.cgi>, télécharger la version 6.0 en vigueur (>= 6.0.35) sous forme de binaires zippés (distribution « Core »). Choisir la version correspondant à votre OS et votre processeur (ex : « 32-bit Windows zip »).
- Dézipper le contenu du fichier dans le répertoire de votre choix.

2.3 Apache HTTP Server 2.2

2.3.1 Installation

- Depuis l'url <http://httpd.apache.org/download.cgi#apache22>, choisir la version correspondant à votre système d'exploitation. Pour Windows, choisir **other files**, puis le répertoire **binaires/win32** et télécharger le fichier **httpd-2.2.XX-win32-x86-openssl-0.9.8t.msi**.
- Lancer l'installation, accepter les conditions d'utilisation et cliquer sur **Next**
- Choisir le port d'écoute 80 et remplir le nom de serveur, de domaine et l'adresse de l'administrateur :

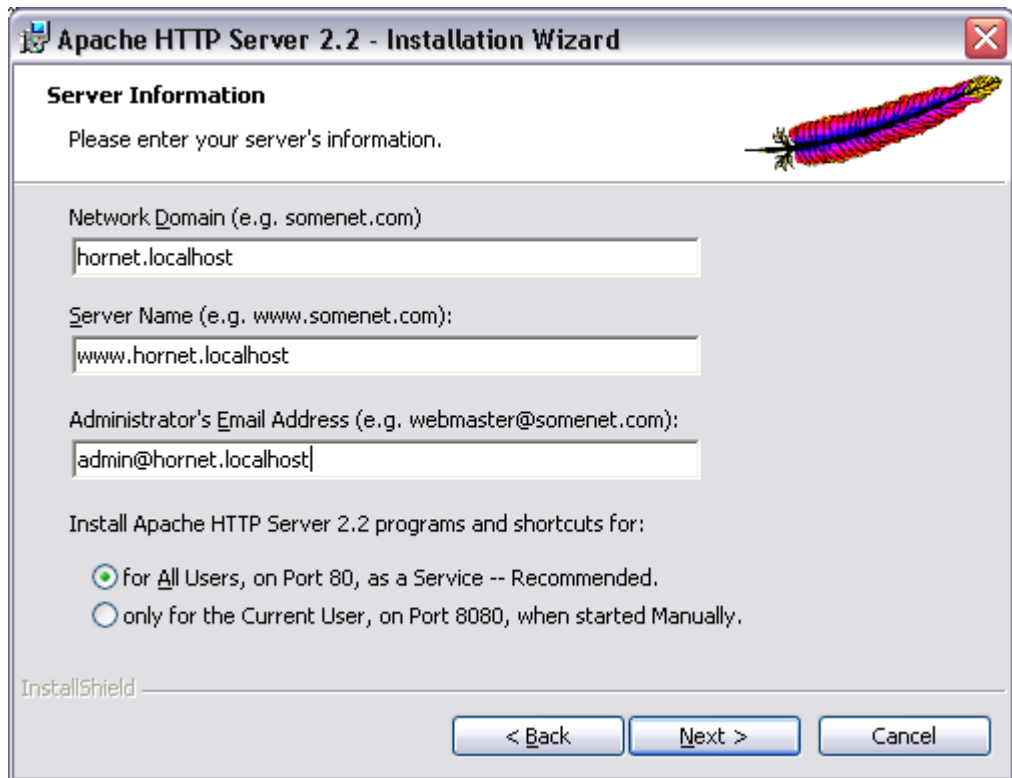


Figure 1 : Configuration lors de l'installation

- Laisser l'option **Typical** pour le type d'installation :

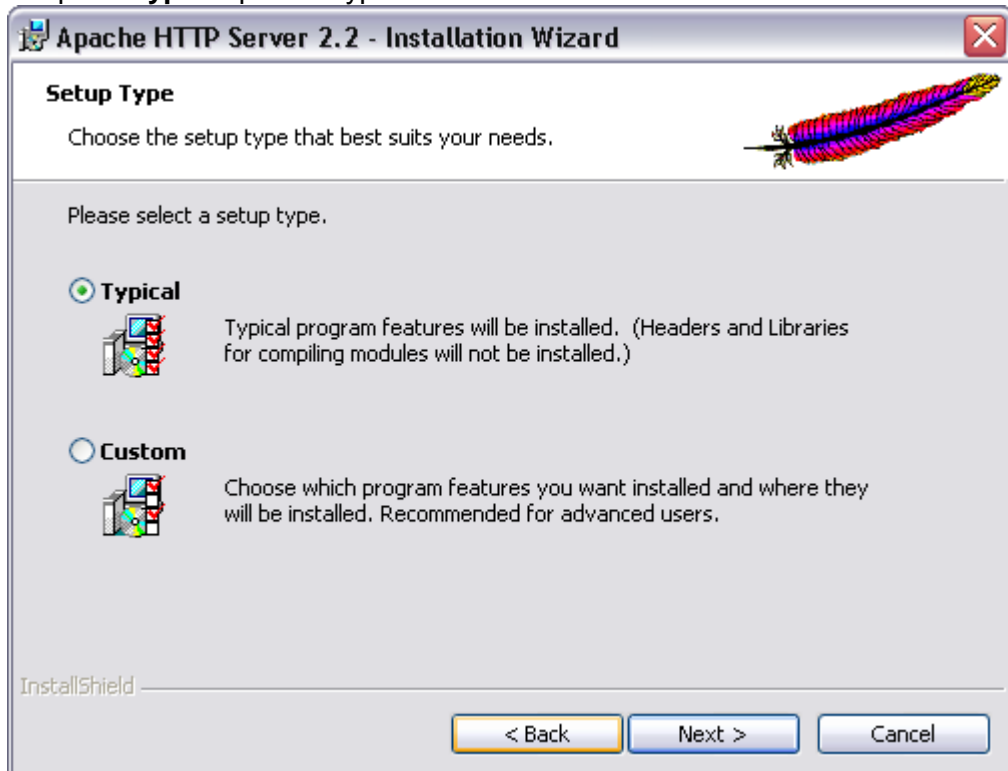


Figure 2 : Choix du type d'installation

- Choisir un répertoire et cliquer sur **Next** puis **Install**.
- Cliquer sur **Finish**

Tester l'installation en ouvrant la page <http://localhost>. La page par défaut doit s'afficher :

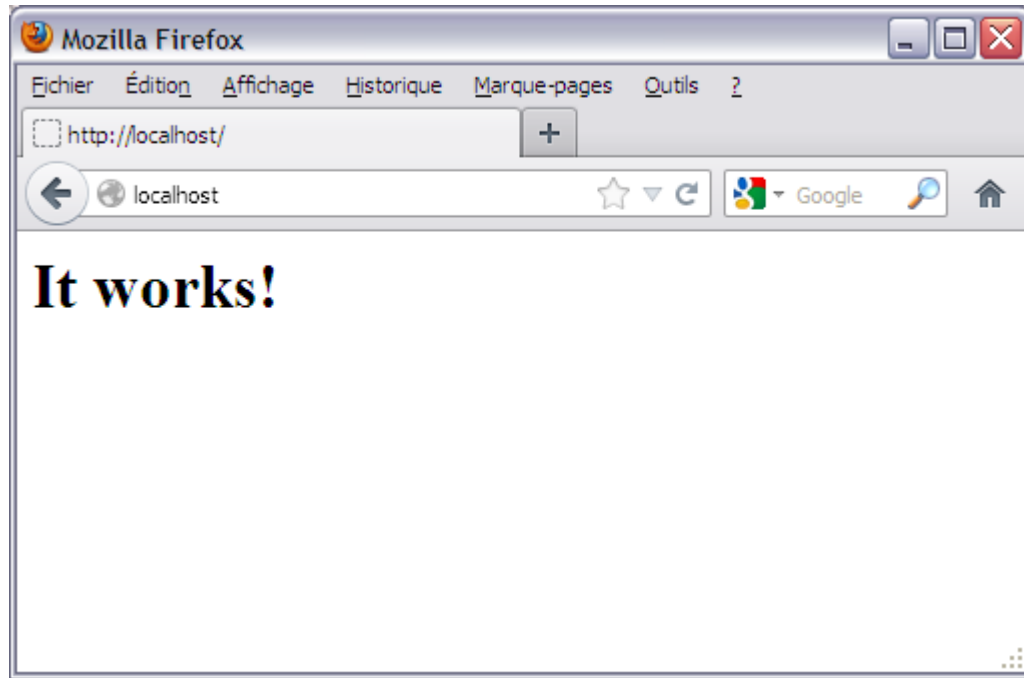


Figure 3 : Page de test du serveur HTTP

2.3.2 Configuration

2.3.2.1 Fichier à modifier

La configuration du serveur HTTP Apache est stockée dans le fichier situé sous `/conf/httpd.conf` dans le répertoire d'installation.

2.3.2.2 Activation des modules

- Ouvrir le fichier **httpd.conf**
- Dans la section relative aux modules (directives Apache « LoadModule ») décommenter les lignes suivantes :

```
LoadModule alias_module modules/mod_alias.so
...
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
...
LoadModule rewrite_module modules/mod_rewrite.so
```

2.3.2.3 Règles de réécriture

- Les règles de réécriture doivent être activées en ajoutant la directive suivante :

```
# Règles de réécriture
RewriteEngine on
```

2.3.2.4 Configuration du connecteur Tomcat

- Ajouter un balancer, utilisant le connecteur AJP du serveur Tomcat, qui pourra être référencé dans des directive ProxyPass pour la redirection serveur HTTP/serveur d'application :

```
# Balancers
<Proxy balancer://balancer4>
    BalancerMember ajp://127.0.0.1:8009 route=jvm1 retry=30 keepalive=on ttl=10 loadfactor=3
flushpackets=auto timeout=300
    Order deny,allow
    Allow from all
</Proxy>
```


2.3.2.5 Exemple de configuration pour l'application HornetTemplate

- Afin d'associer un format d'URL à un balancer, on utilise la directive ProxyPass :

```
ProxyPass /hornettemplate/dyn/ balancer://balancer4/hornettemplate/dyn/  
stickysession=JSESSIONID|jsessionid nofailover=on  
ProxyPass /hornettemplate/index.html balancer://balancer4/hornettemplate/dyn/protected/accueil.html  
stickysession=JSESSIONID|jsessionid nofailover=on
```

Dans le cas présent, les URLs du type `http://serveur.extension/hornettemplate/dyn/*` seront routées vers le balancer configuré pour le connecteur Tomcat avec le contexte `/hornettemplate/dyn/*`.

Dit plus simplement, c'est le serveur d'application Tomcat qui prend en charge la construction des pages à l'emplacement `/hornettemplate/dyn/*`.

- Les ressources statiques (images, fichiers JavaScript et CSS) nécessitent la configuration d'un alias :

```
Alias /hornettemplate/static "D:/devjava/workspace/hornettemplate/WebContent/static"  
<Directory "D:/devjava/workspace/hornettemplate/WebContent/static ">  
    AllowOverride None  
    Order allow,deny  
    allow from all  
</Directory>
```

Cet élément de configuration indique à Apache où aller chercher les fichiers accessibles depuis l'URL `http://serveur.extension/hornettemplate/static/`.

Au final, l'application `hornettemplate` générée à partir du gabarit (cf. Guide de Création d'un Projet) nécessite la configuration suivante :

```
## hornettemplate default ##  
Alias /hornettemplate/static D:/devjava/workspace/hornettemplate/WebContent/static  
<Directory "D:/devjava/workspace/hornettemplate/WebContent/static ">  
    AllowOverride None  
    Order allow,deny  
    allow from all  
</Directory>  
RewriteRule (^/hornettemplate/)(?!dyn)(.*) (;jsessionid=.*) $1$2 [PT]  
ProxyPass /hornettemplate/dyn/ balancer://balancer4/hornettemplate/dyn/  
stickysession=JSESSIONID|jsessionid nofailover=on  
ProxyPass /hornettemplate/authentication/ balancer://balancer4/hornettemplate/authentication/  
stickysession=JSESSIONID|jsessionid nofailover=on  
ProxyPass /hornettemplate/j_spring_security_check  
balancer://balancer4/hornettemplate/j_spring_security_check stickysession=JSESSIONID|jsessionid  
nofailover=on  
ProxyPass /hornettemplate/j_spring_security_login  
balancer://balancer4/hornettemplate/j_spring_security_login stickysession=JSESSIONID|jsessionid  
nofailover=on  
ProxyPass /hornettemplate/j_spring_security_logout  
balancer://balancer4/hornettemplate/j_spring_security_logout stickysession=JSESSIONID|jsessionid  
nofailover=on  
ProxyPass /hornettemplate/index.html balancer://balancer4/hornettemplate/dyn/protected/accueil.html  
stickysession=JSESSIONID|jsessionid nofailover=on
```

3 IDE Eclipse

3.1 Installation

- Depuis l'URL <http://www.eclipse.org/downloads/packages/release/kepler/sr1>, télécharger le package **Eclipse IDE for Java EE Developers** (fichier eclipse-jee-kepler-SR1-win32.zip)
- Extraire le fichier dans le répertoire de votre choix.

3.2 Configuration

3.2.1 Lancement

Afin de limiter les risques d'utiliser un mauvais encodage des fichiers ou une mauvaise version de Java, il est recommandé de lancer Eclipse avec les paramètres suivants :

```
-vm C:\Progra~1\Java\jdk1.6.0_26\bin\ -Dfile.encoding=UTF-8
```

-vm indique l'emplacement de la machine virtuelle Java à utiliser pour l'exécution d'Eclipse

-Dfile.encoding permet de positionner la variable système Java « file.encoding » (utilisée par défaut) avec l'encodage UTF-8

Sous Windows, ces paramètres peuvent être ajoutés en modifiant la propriété « cible » du raccourci correspondant.

Il est également possible de modifier le fichier eclipse.ini directement.

Exemple de configuration :

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.0.v20130327-1440.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_1.1.200.v20130807-1835
-product
org.eclipse.epp.package.jee.product
--launcher.defaultAction
openFile
--launcher.XXMaxPermSize
256M
-vm
C:\Program Files\Java\jdk1.6.0_45\bin
-showsplash
org.eclipse.platform
--launcher.XXMaxPermSize
256m
--launcher.defaultAction
openFile
--launcher.appendVmargs
-vmargs
-Dosgi.requiredJavaVersion=1.6
-Xms40m
-Xmx512m
-Dfile.encoding=UTF-8
```

Note :

L'argument « -vm » doit absolument être avant l'option « -vmargs »

Les clés/valeurs doivent être sur deux lignes pour l'argument « -vm ».

L'argument « -Dfile.encoding » est ajouté après « -vmargs ».

3.2.2 Création du workspace

Démarrer Eclipse et choisir un emplacement pour le workspace (créer un nouveau dossier auparavant si nécessaire) :

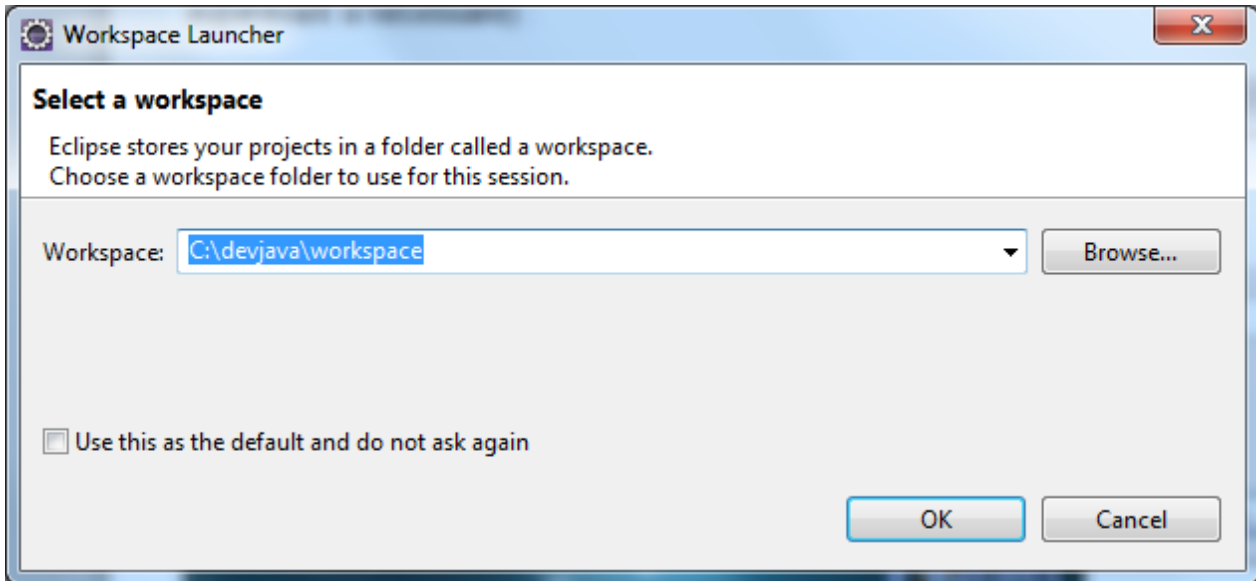


Figure 4 : Sélection du workspace

3.2.3 Paramétrage du JDK

- Choisir le menu **Window** → **Preferences** puis sélectionner **Java** → **Installed JREs**
- Cliquer sur **Add**
- Choisir **Standard VM** puis **Next**
- Renseigner le champ **Home** en sélectionnant le répertoire d'installation du JDK :

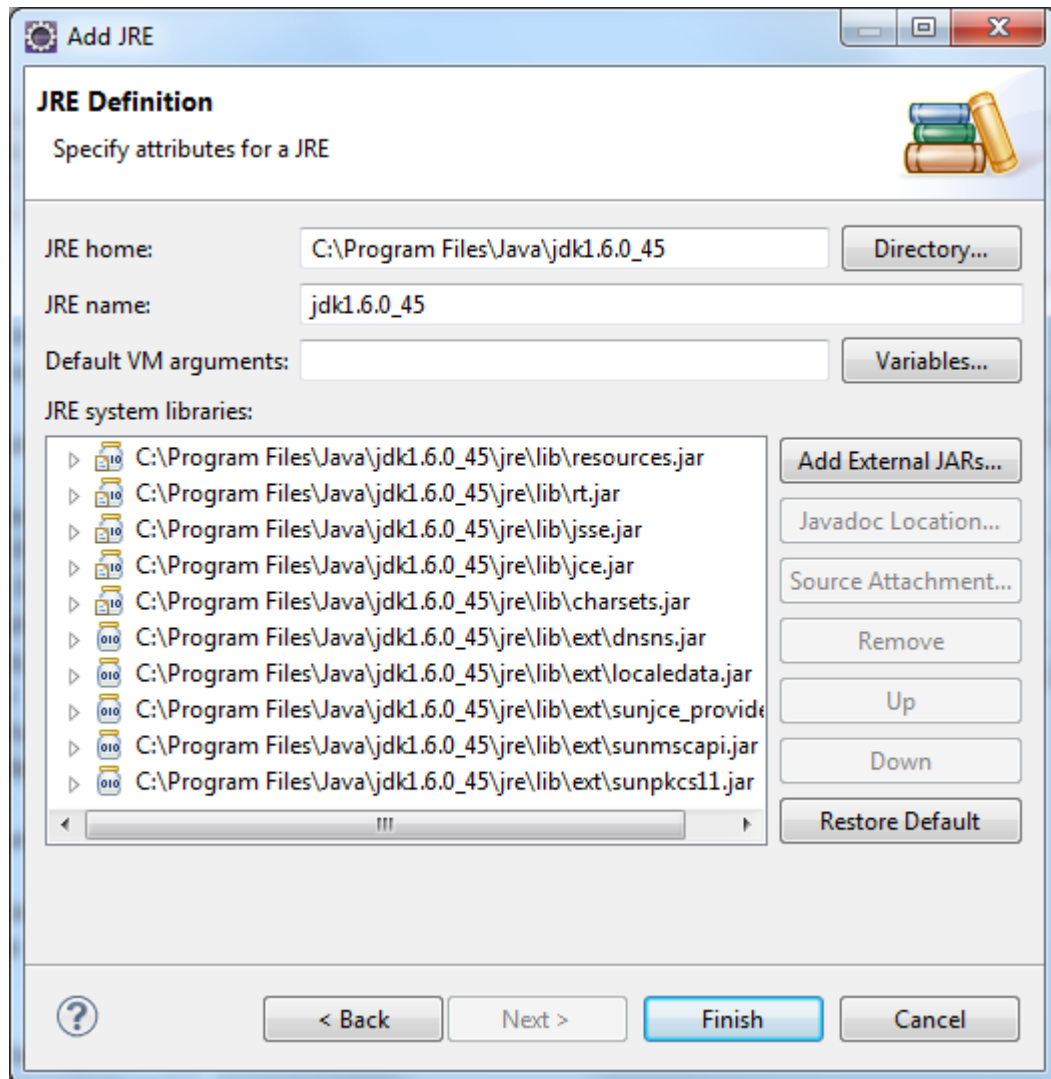


Figure 5 : Définition d'un nouveau JRE

- Cliquer sur **Finish** et cocher la case jdk1.6.0_XX :

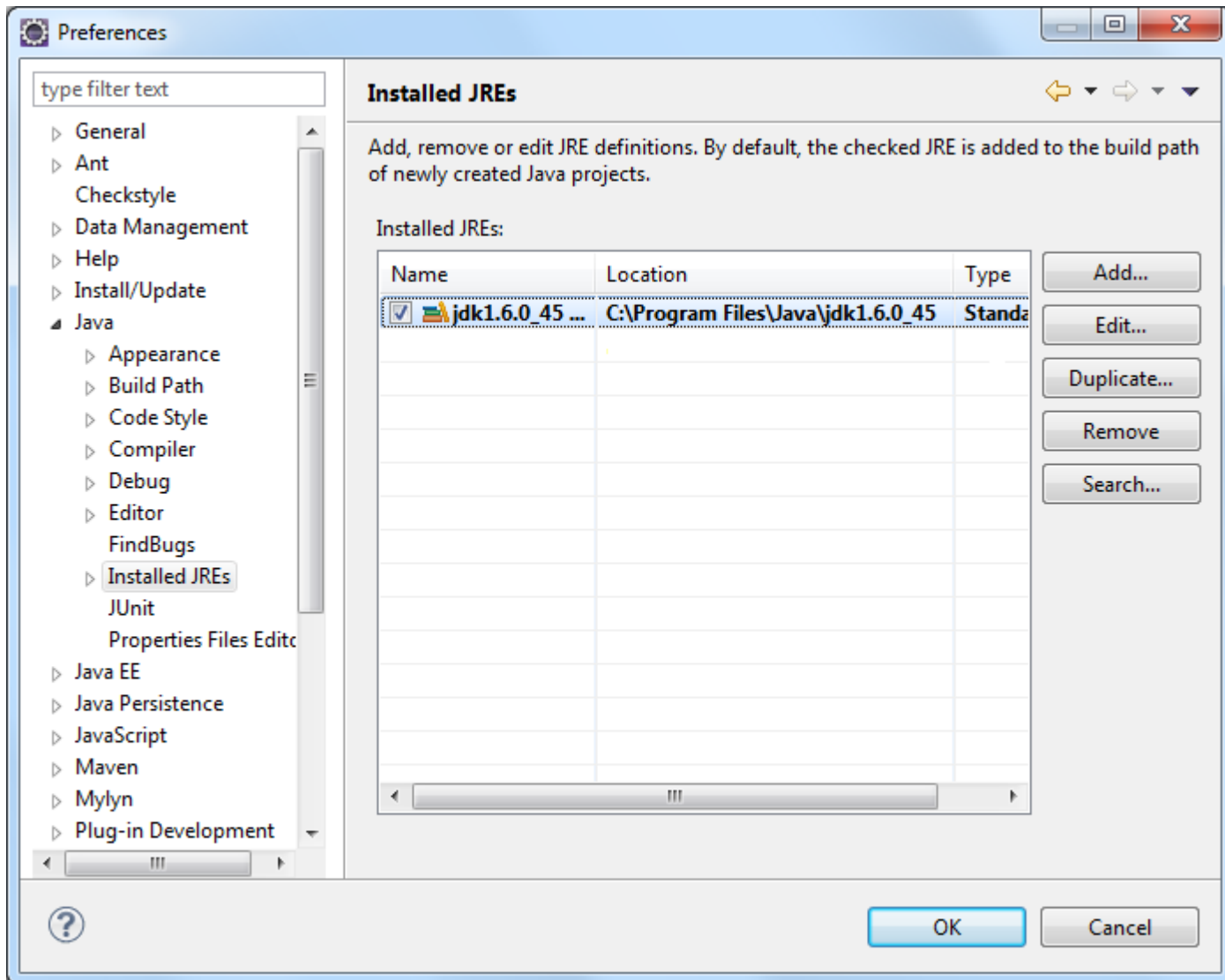


Figure 6 : Choix du JDK

- Valider en cliquant sur **OK**

3.2.4 Environnement d'exécution Tomcat

Les projets Web à partir d'Hornet 3, comme l'AppliTutoriel, référencent un environnement d'exécution Tomcat 6.0 permettant de lancer un serveur de test depuis Eclipse. Pour le configurer :

- Choisir le menu **Window** → **Preferences** puis sélectionner **Server** → **Runtime Environnements**
- Cliquer sur **Add**
- Choisir **Apache Tomcat 6.0** puis **Next**
- Choisir le nom par défaut **Apache Tomcat v6.0**, renseigner le champ **Tomcat installation directory** en sélectionnant le répertoire d'installation de Tomcat puis choisir **jdk1.6.0_XX** pour le champ **JRE** :

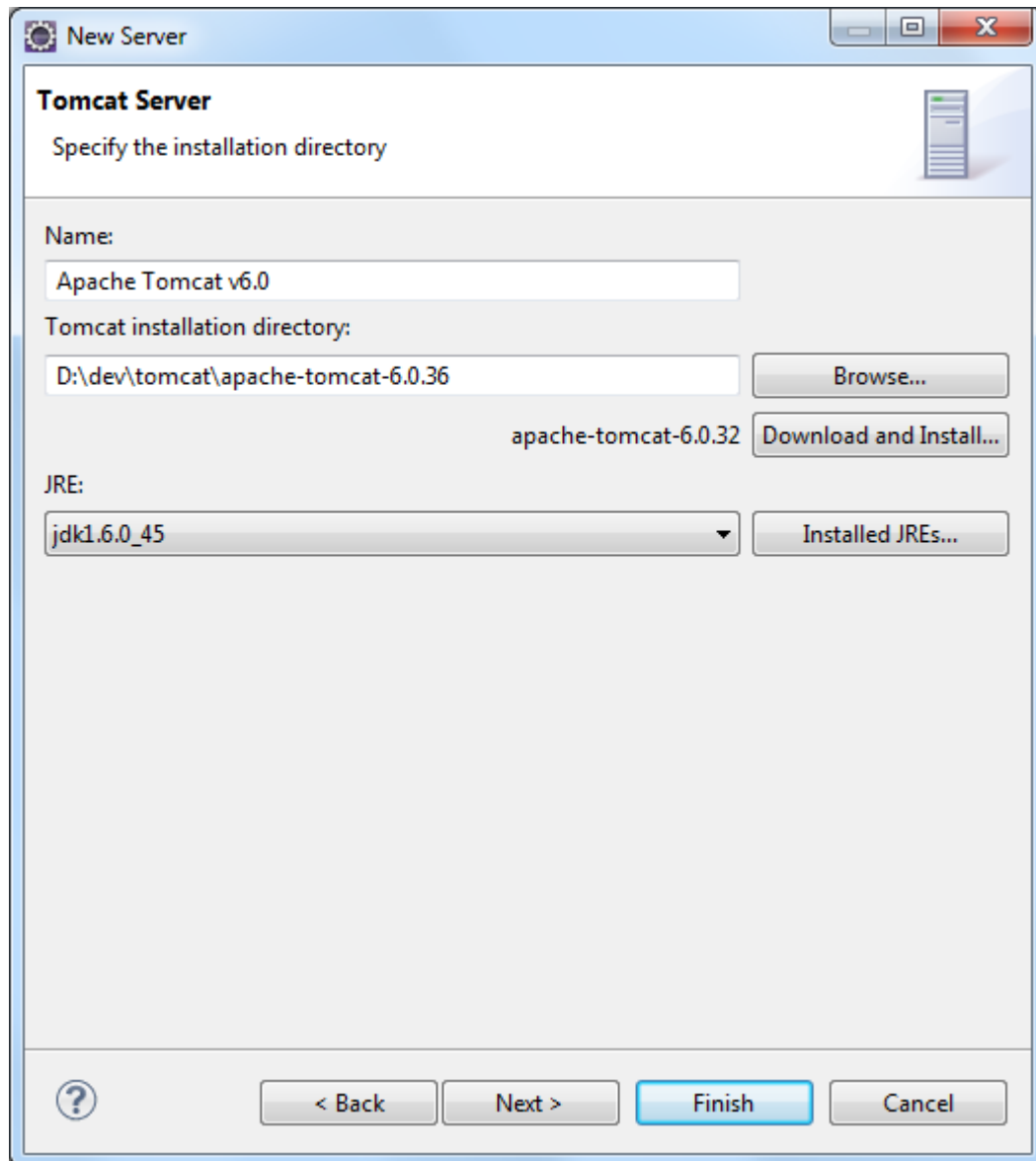


Figure 7 : Déclaration du serveur Tomcat

- Cliquer sur **Finish** puis **Ok**

3.2.5 Encoding (et format de fin de ligne)

⇒ **Important** : Il existe deux paramètres intervenant dans l'encodage d'un fichier. Le premier, pour les formats le permettant tels que ceux dérivant du XML, est indiqué en début de document et est visible par l'utilisateur lors de son édition. Le second se paramètre dans l'éditeur utilisé et n'est pas visible directement.

Eclipse permet de spécifier le format d'un fichier : sous Windows, vérifier que l'encodage défini pour le workspace est bien l'UTF-8 dans la fenêtre **Window** → **Preferences**, puis rubrique **General** → **Workspace**, option **Text File Encoding**.

De la même manière, pour forcer le format de fin de ligne en mode Unix, sur le même écran, vérifier l'option **New text file line delimiter**, à positionner sur Unix.

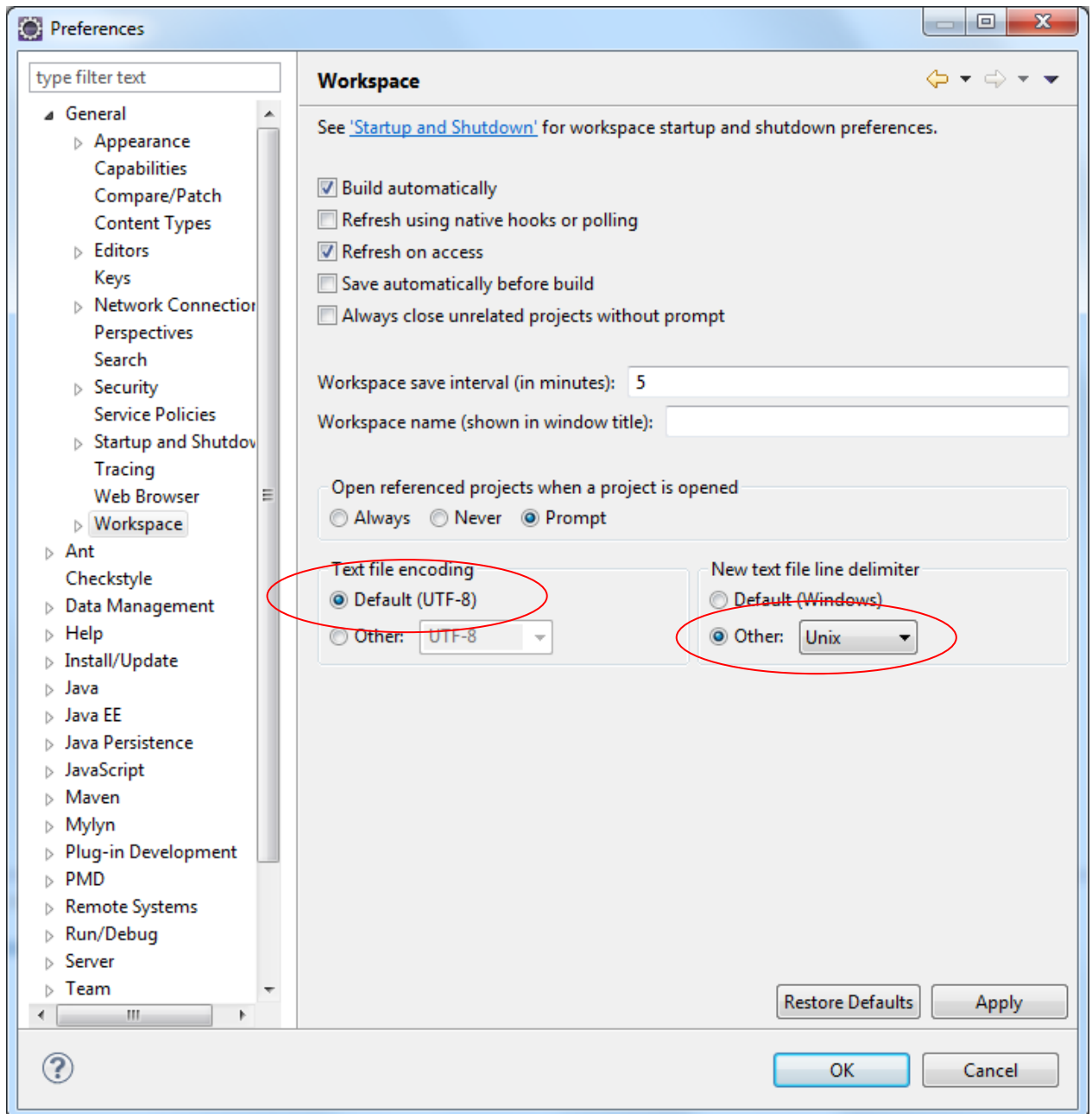


Figure 8 : Fenêtre de préférences générales d'Eclipse

Il est également possible d'éditer ce paramètre pour un fichier spécifique ou un dossier en ouvrant la boîte de dialogue des propriétés de ce fichier.

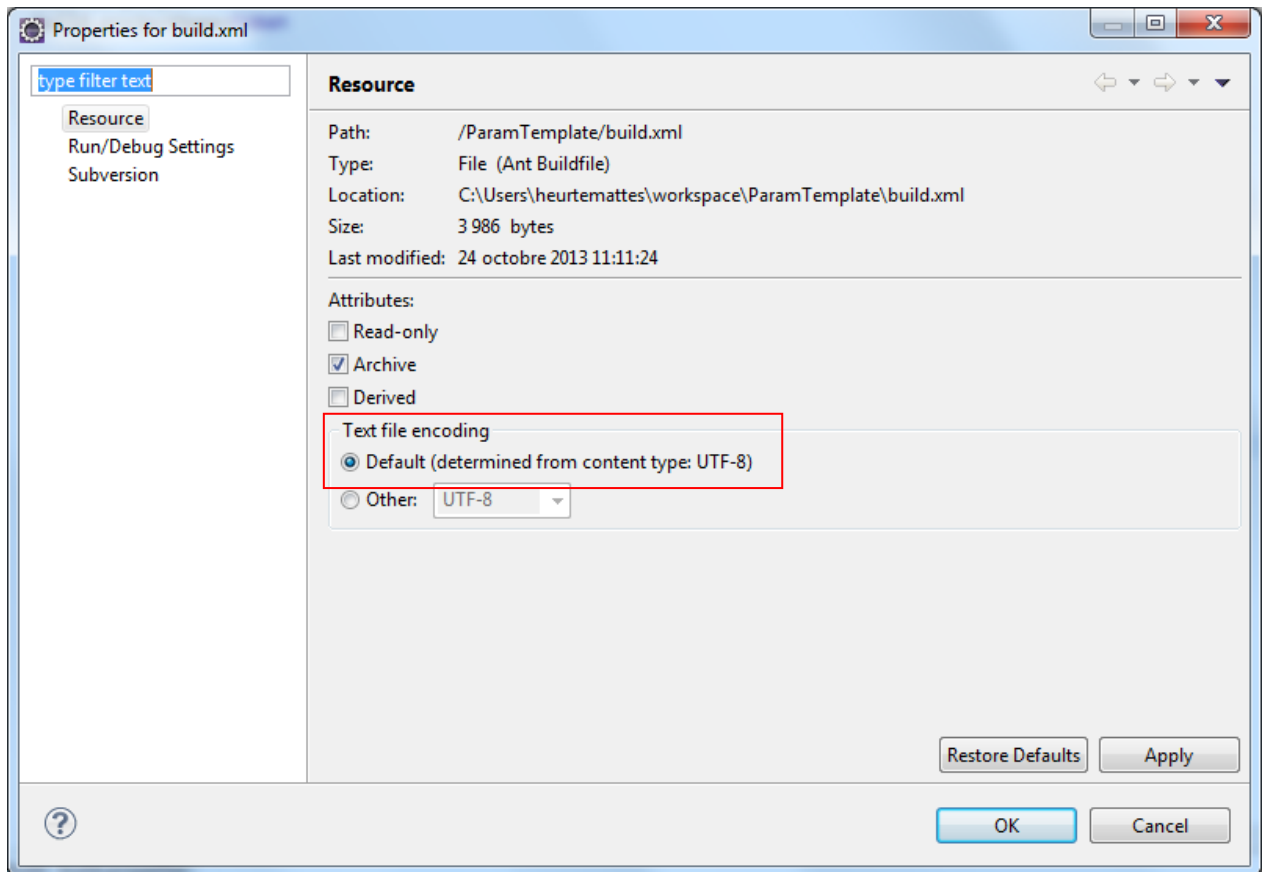


Figure 9 : Fenêtre de préférences d'un fichier dans Eclipse

Remarques :

- L'ISO-8859-15 n'apparait pas dans cette liste. Il est cependant sélectionné automatiquement si Eclipse l'a détecté dans une balise du document, par exemple avec l'entête suivant pour les fichiers HTML :

```
...
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-15">
...
```

- Les fichiers properties sont systématiquement écrits en ISO-8859-1. C'est une norme définie par Java. Les caractères spéciaux doivent alors être échappés en utilisant une notation de la forme « \uXXXX ».

Il est fortement conseillé d'utiliser Eclipse pour éditer des pages web contenant des caractères spéciaux en UTF-8, et pas des éditeurs comme le bloc notes qui ne proposent pas assez d'options d'encodage. Des informations différentes entre l'encodage du fichier et l'encodage indiqué dans une balise risquent d'aboutir à des caractères spéciaux corrompus.

3.2.6 Apache Ant

Eclipse Kepler intègre directement un environnement d'exécution Ant en version 1.8.2. Cet élément ne nécessite pas de configuration particulière.

3.3 Paramétrage Hornet

3.3.1 Présentation

Afin d'améliorer la qualité du code en termes de présentation et de respect des bonnes pratiques, en particulier des normes de développement Java, trois plugins complémentaires sont utilisés pour le développement Hornet :

- PMD
- FindBugs
- Checkstyle : Outil de développement qui, contrairement aux deux précédents, permet la vérification du code en cours de frappe.

De plus, afin d'assurer l'homogénéité des règles de développement, des éléments additionnels de formatage sont à intégrer dans la configuration d'Eclipse.

3.3.2 Gestionnaire d'installation

Eclipse gère ses plugins depuis des dépôts appelés « Update Sites ». Il en existe un (officiel) par distribution d'Eclipse mais certains plugins sont disponibles uniquement depuis des dépôts spécifiques.

L'installation de nouveaux plugins est accessible depuis le menu **Help** → **Install New Software** :

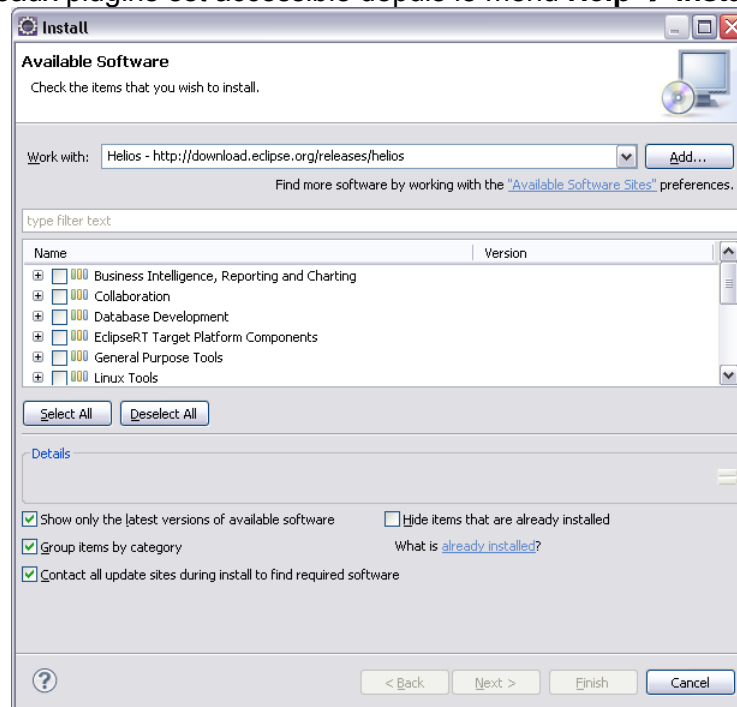


Figure 10 : Gestionnaire d'installation de plugins Eclipse

Cette méthode permet la récupération des dernières versions des plugins. Cependant, en cas d'une version particulière d'un plugin, la démarche sera décrite dans la section concernée.

3.3.3 PMD

3.3.3.1 Installation

- Dans le champ « **Work With** » du gestionnaire de plugins, indiquer l'url <http://pmd.sourceforge.net/eclipse/>, valider et attendre le chargement.
- Cocher **PMD for Eclipse 4** :
- Cliquer sur « **Next** » deux fois, accepter le contrat de licence et cliquer sur « **Finish** ».

- L'installation se lance et stoppe sur un warning indiquant que les jars ne sont pas signés. Valider.
- Un autre prompt demande le redémarrage d'Eclipse. Choisir « **Restart Now** »

3.3.3.2 Configuration

Dans le cas d'Hornet, ce fichier est disponible dans la section « Fichiers / PMD » de la page Adullact Hornet (http://adullact.net/frs/?group_id=761).

- Récupérer le fichier sur le disque : **HORNET-PMD-4.0.1-way-1.X.xml**
- Accéder à l'écran de gestion de préférences d'Eclipse, via le menu **Windows** → **Preferences**.
- Accéder à la configuration « **PMD** → **Configuration des règles** »
- Sélectionner toute les règles avec « **Ctrl + A** » et puis « **Suppr** » (sur le clavier)

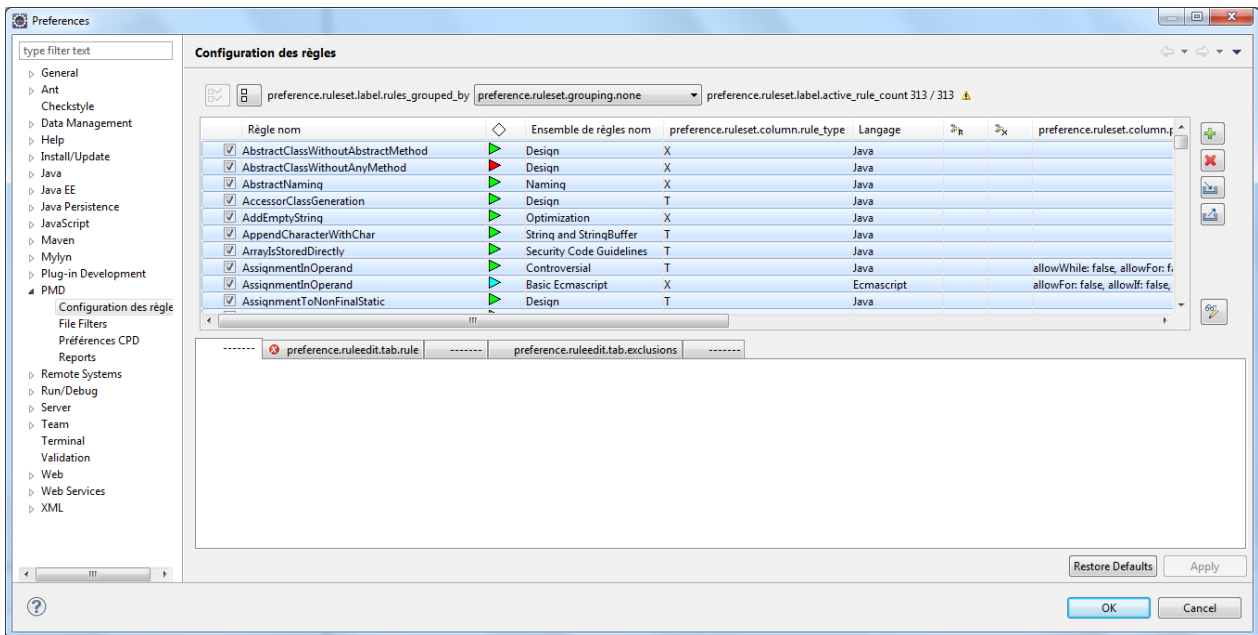


Figure 11 : Suppression des règles PMD

- Cliquer sur l'icône « **Import** » et sélectionner le fichier téléchargé

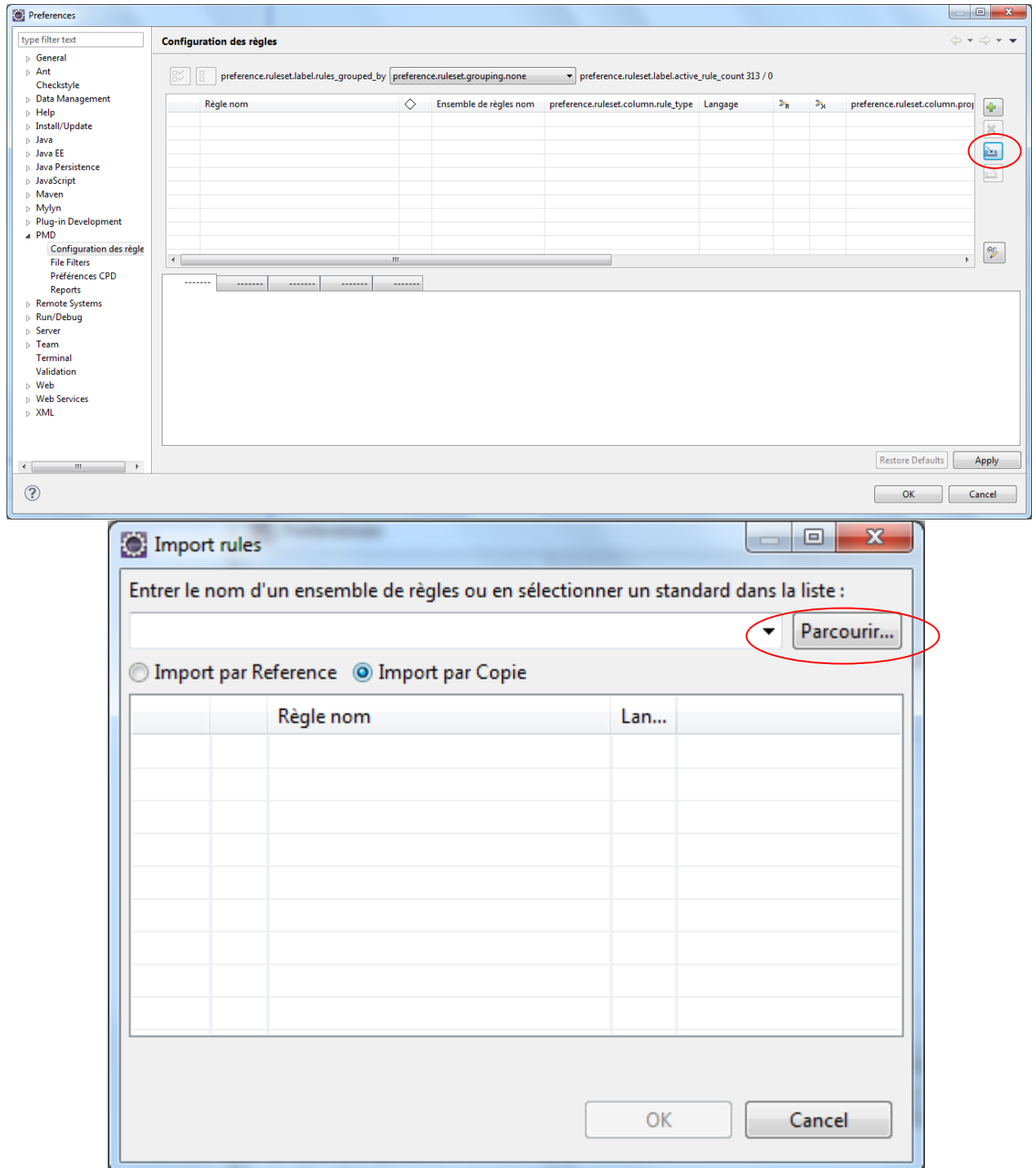


Figure 11 : Import des règles

3.3.4 FindBugs

- Dans le champ « **Work With** » du gestionnaire de plugins, indiquer l'URL <http://findbugs.cs.umd.edu/eclipse>, valider et attendre le chargement.
- Cocher **FindBugs Features** :
- Cliquer sur **Next** deux fois, accepter le contrat de licence et cliquer sur **Finish**.
- L'installation se lance et stoppe sur un warning indiquant que les jars ne sont pas signés. Valider.
- Un autre prompt demande le redémarrage d'Eclipse. Choisir **Restart Now**

3.3.5 Checkstyle

3.3.5.1 Installation

- Dans le champ « **Work With** » du gestionnaire de plugins, indiquer l'url <http://eclipse-cs.sourceforge.net/update/>, valider et attendre le chargement.
- Cliquer sur « **Next** » deux fois, accepter le contrat de licence et cliquer sur « **Finish** ».
- L'installation se lance et stoppe sur un warning indiquant que les jars ne sont pas signés. Valider.
- Un autre prompt demande le redémarrage d'éclipse. Choisir « **Restart Now** »

3.3.5.2 Configuration

Pour définir les règles de validation du code source à contrôler, le plugin Checkstyle utilise un fichier de configuration pouvant être stocké sur un serveur central (partage réseau, serveur http ...).

Dans le cas d'Hornet, ce fichier est disponible dans la section « Fichiers / Checkstyle » de la page Adullact Hornet (http://adullact.net/frs/?group_id=761).

- Télécharger ce fichier sur le disque
- Sous Eclipse cliquer sur **Window** → **Préférences** puis **CheckStyle**

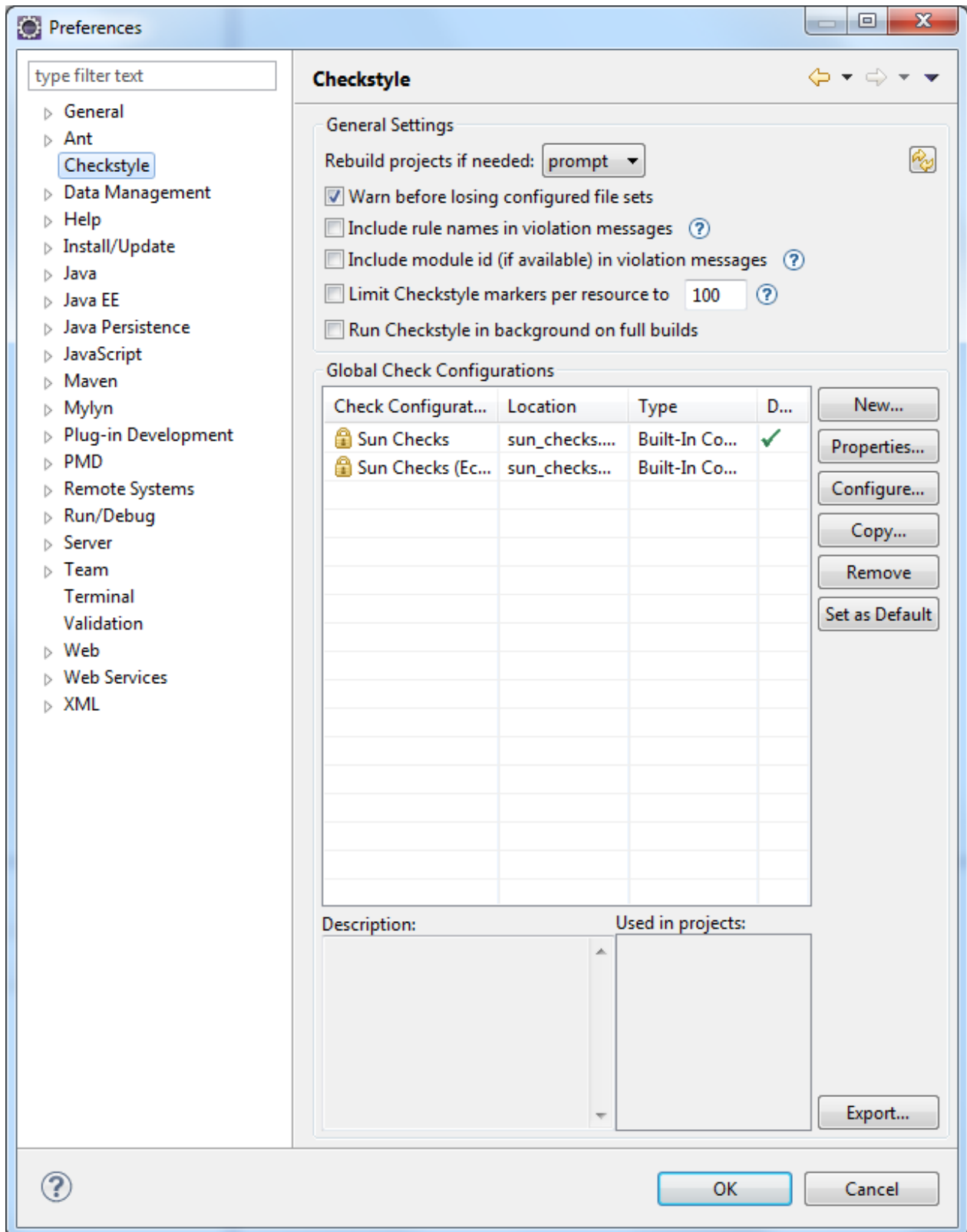


Figure 12 : Plugin Checkstyle : Configuration

- Cliquer sur **New** et choisir « **Configuration externe** ». Nommer la configuration
- Cliquer sur **Browse ...** et sélectionner le fichier XML précédemment sauvegardé :

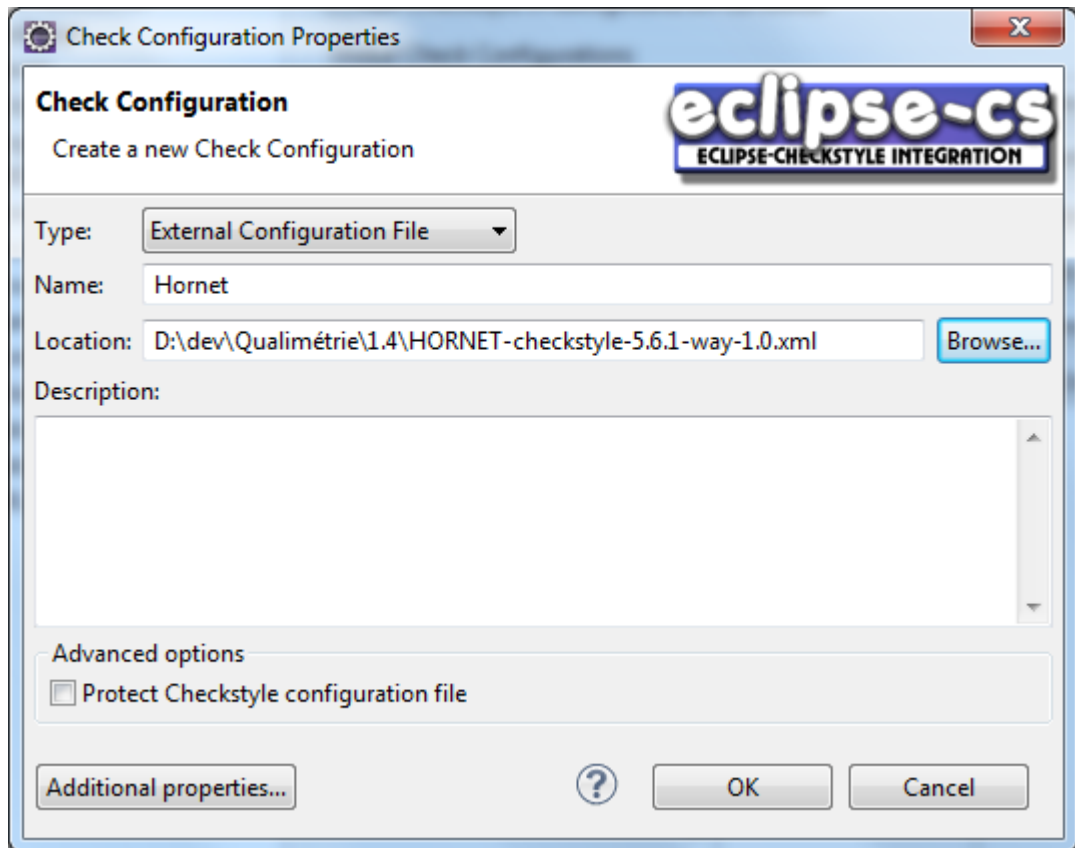


Figure 13 : Plugin Checkstyle : Référencement de la configuration

- Cocher la case de protection du fichier de configuration puis valider.
- Cocher la case d'inclusion des noms de règles puis valider.

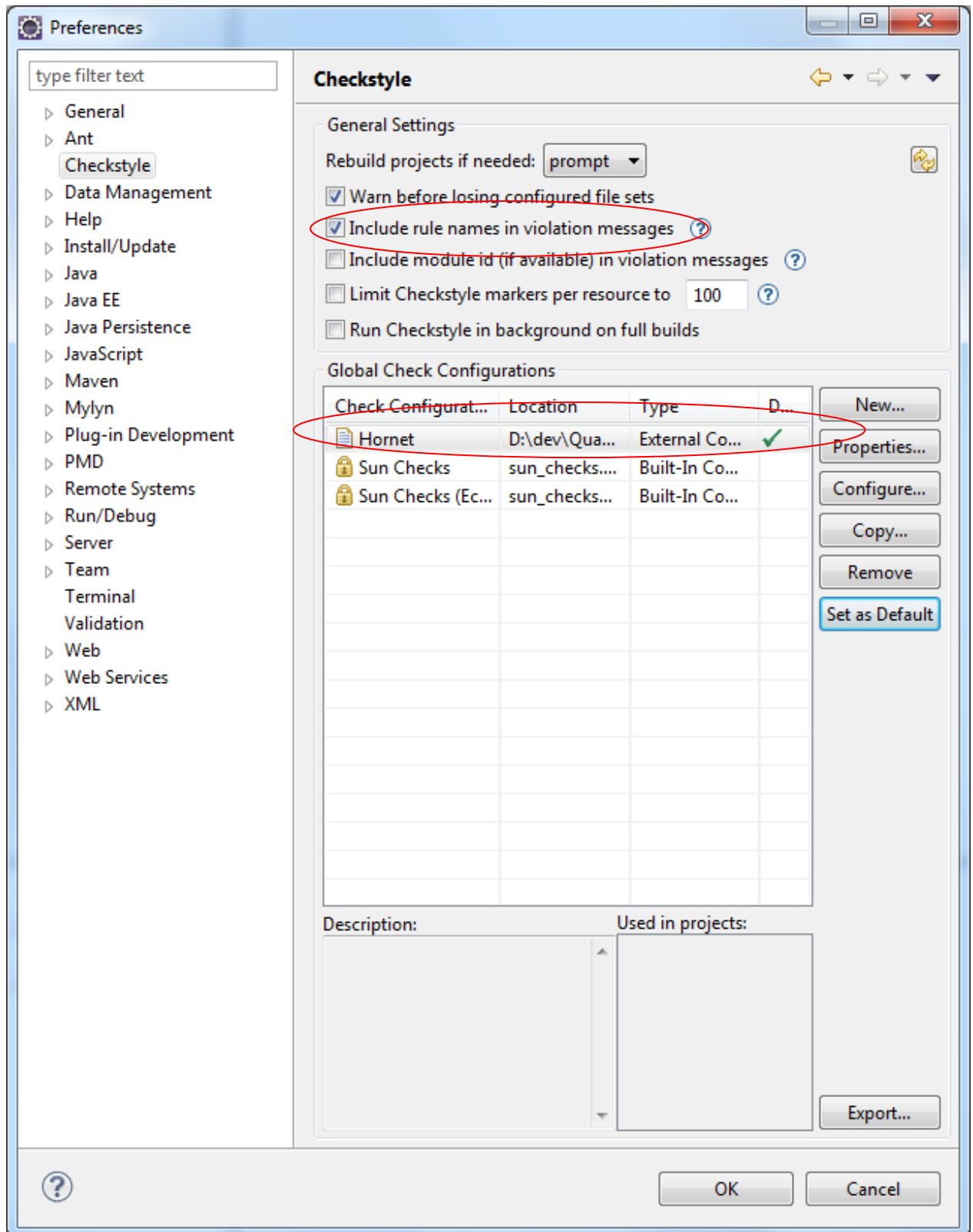


Figure 14 : Plugin Checkstyle : Etape finale de configuration

3.3.5.3 Paramétrage des projets

Lors du développement de projets Hornet, il est bon de vérifier la bonne application de la configuration Checkstyle en vérifiant les propriétés du projet

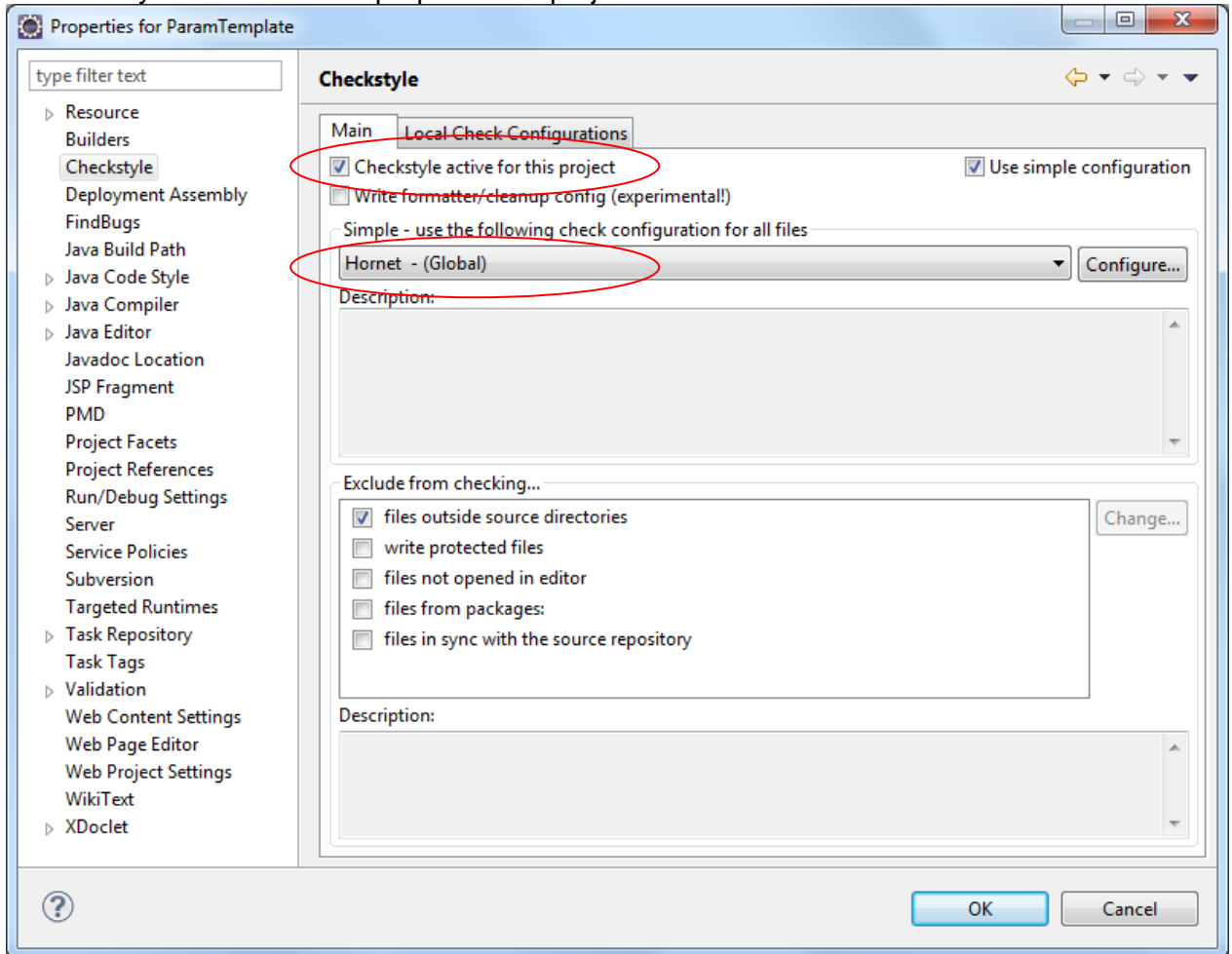


Figure 15 : Propriété d'un projet : Vérification Checkstyle

3.3.1 Formatage et modèles

En fonction des types de fichiers utilisés dans Eclipse, un formatage de ces derniers est recommandé. L'ensemble de ces modifications sont à apporter dans l'écran de gestion de préférences d'Eclipse, via le menu **Windows** → **Preferences**.

3.3.1.1 Formatage de code Java

- Accéder à l'option de paramétrage **Java** → **Code Style** → **Formatter** et importer un nouveau profil de formatage via le bouton « **Import ...** »

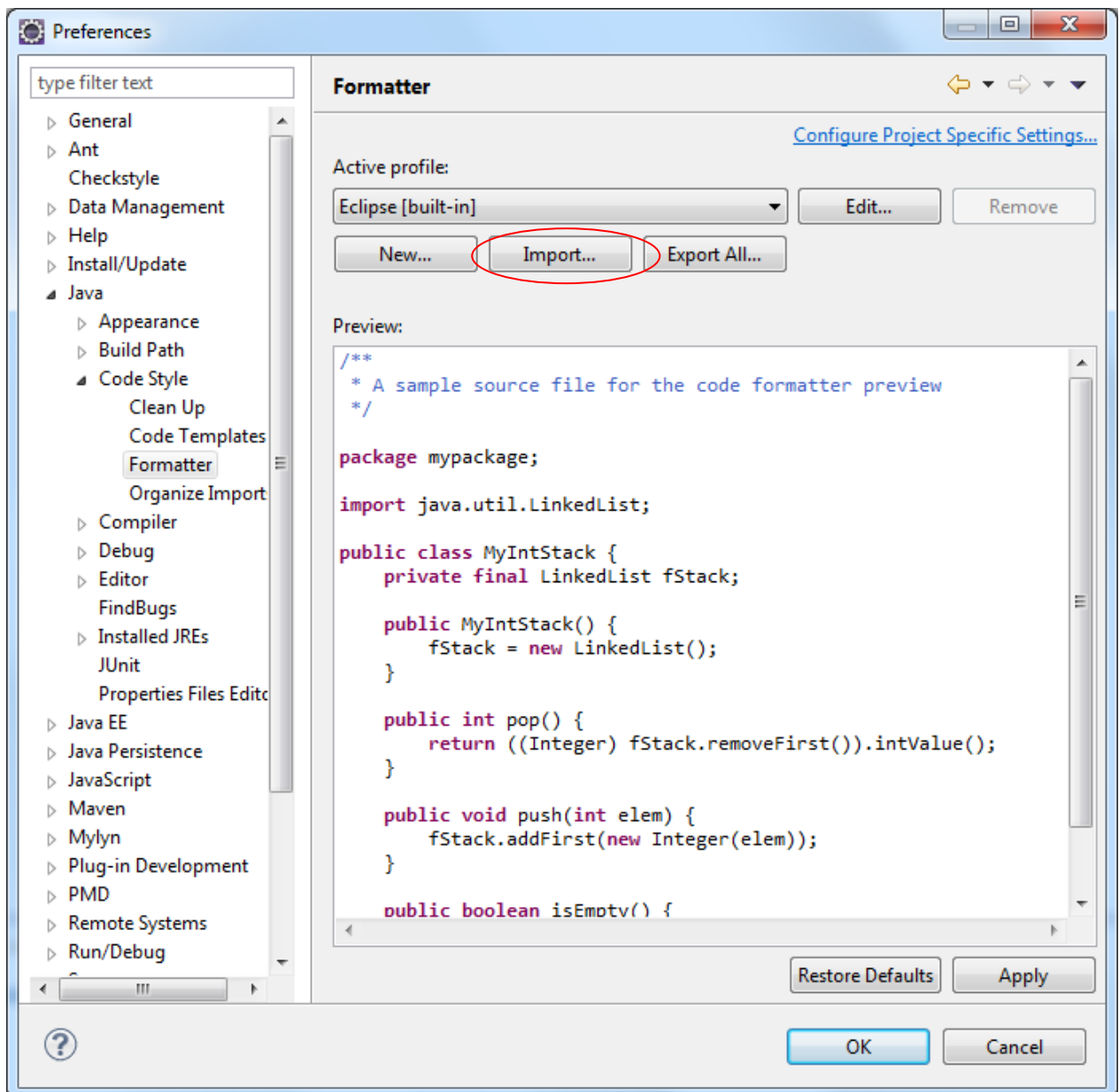


Figure 16 : Formatage Java : Nouveau profil

- Sélectionner le fichier de formatage « **HORNET-eclipse-java-format-X.X.xml** » puis valider.

3.3.1.2 Modèle de code Java

- Accéder à l'option de paramétrage **Java** → **Code Style** → **Code Template** et importer un nouveau modèle de code Java via le bouton « **Import ...** »

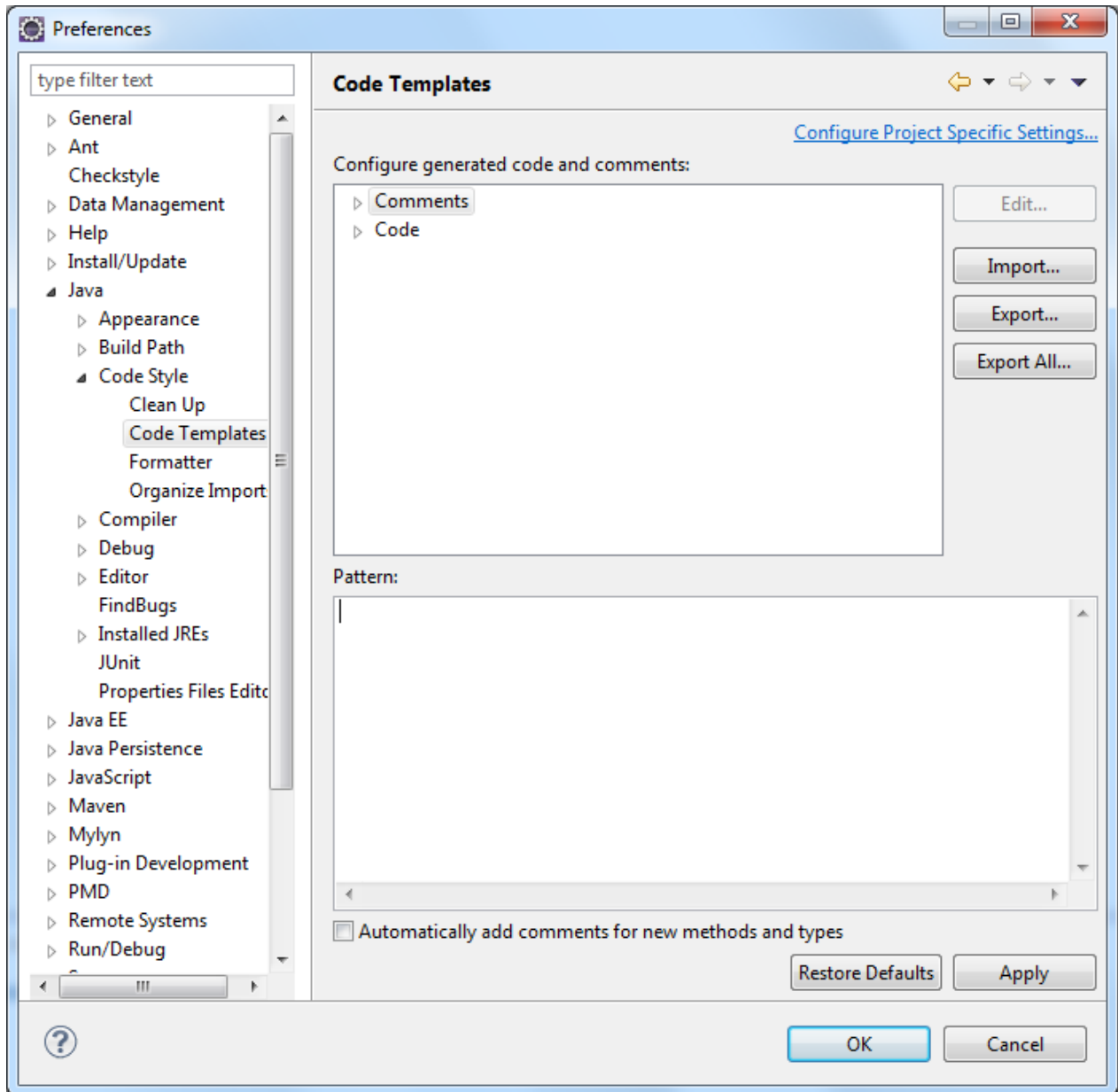


Figure 17 : Modèle Java : Nouvelle configuration

- Sélectionner le fichier de formatage « **HORNET-eclipse-java-codetemplates-X.X.xml** » puis valider.

3.3.1.3 Formatage de code JavaScript

- Accéder à l'option de paramétrage **JavaScript** → **Code Style** → **Formatter** et importer un nouveau profile de formatage via le bouton « **Import ...** »

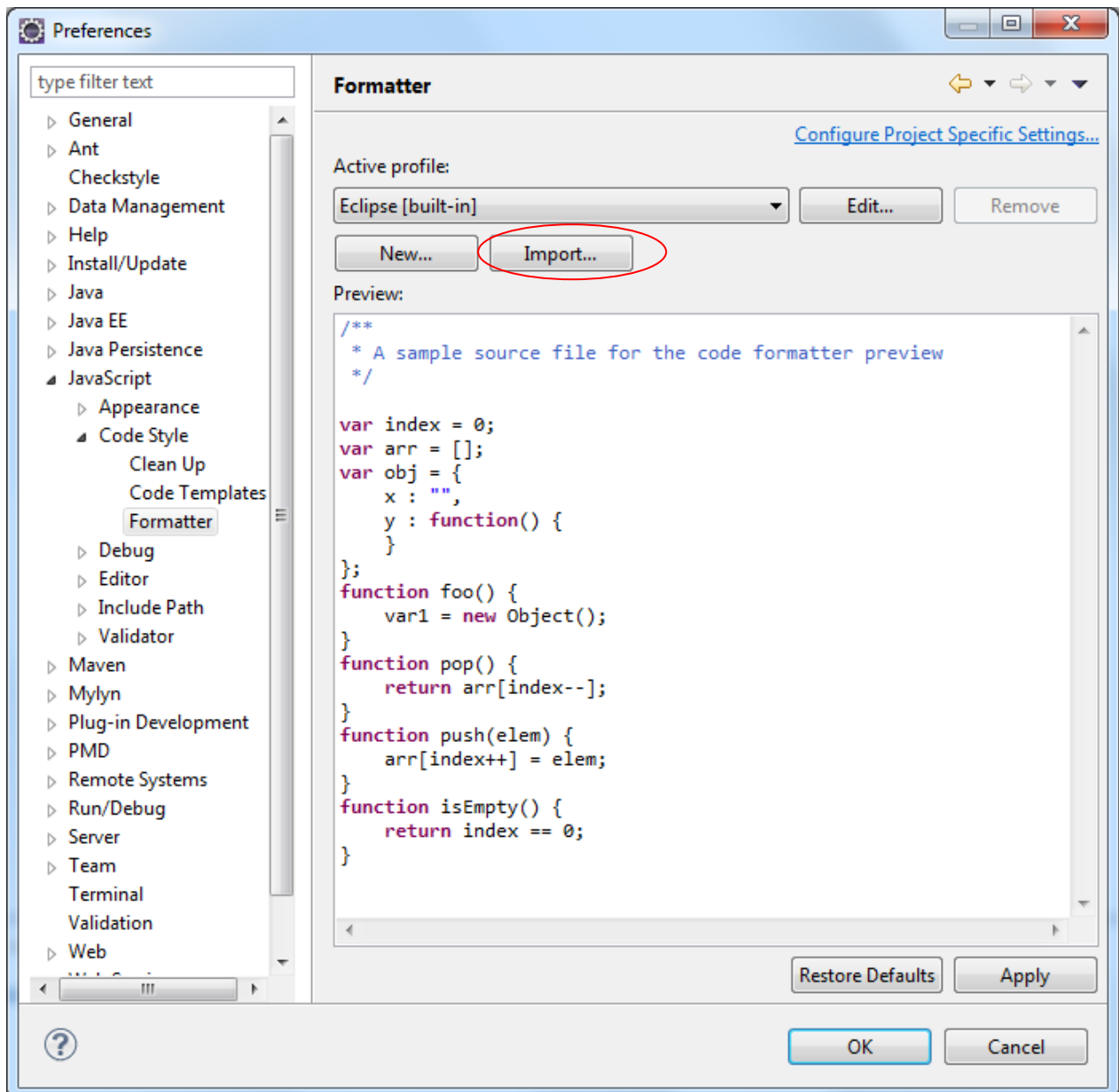


Figure 18 : Formatage Java : Nouveau profile

- Sélectionner le fichier de formatage « **HORNET-eclipse-js-format-X.X.xml** » puis valider.

3.4 Plugins complémentaires

3.4.1 Présentation

En compléments des plugins obligatoires pour les normes de développement Java, le plugin suivant est également recommandé pour faciliter les développements Hornet :

- Subclipse

3.4.2 Subclipse

Subclipse permet une intégration directe de SVN au sein d'Eclipse.

3.4.2.1 Installation

- Accéder à l'interface de gestion des plugins comme référencé au paragraphe 3.3.2.
- Dans le champ **Work With**, indiquer l'url http://subclipse.tigris.org/update_1.8.x, valider et attendre le chargement.
- Cocher la configuration suivante :

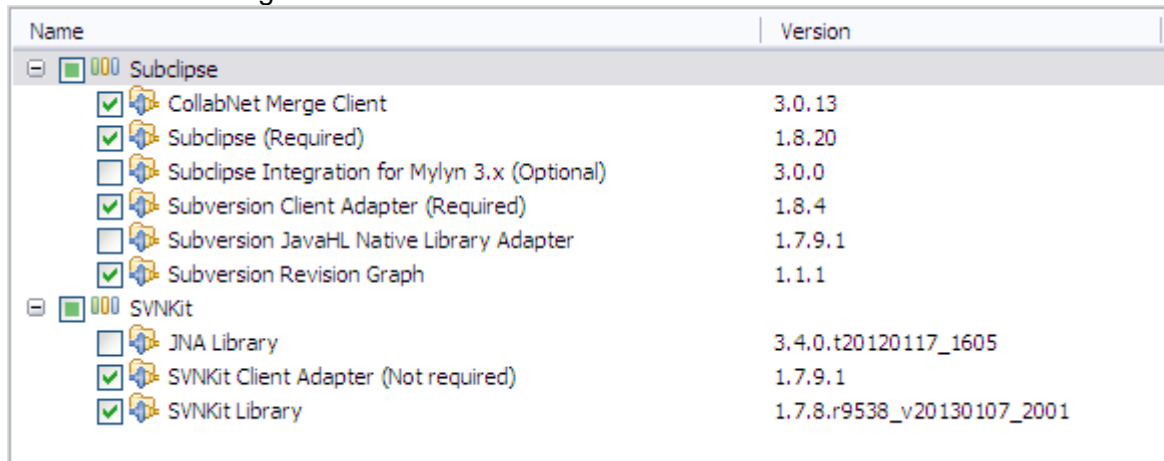


Figure 19 : Configuration Subclipse

- Cliquer sur **Next** deux fois, accepter le contrat de licence et cliquer sur **Finish**.
- L'installation se lance et stoppe sur un warning indiquant que les jars ne sont pas signés. Valider
- Un autre prompt demande le redémarrage d'éclipse. Choisir **Restart Now**

3.4.2.2 Configuration du dépôt Adullact

- Afficher la vue **Svn Repositories** en sélectionnant dans le menu **Window** → **Show view** → **Other ...** puis **Svn** → **Svn Repositories** :

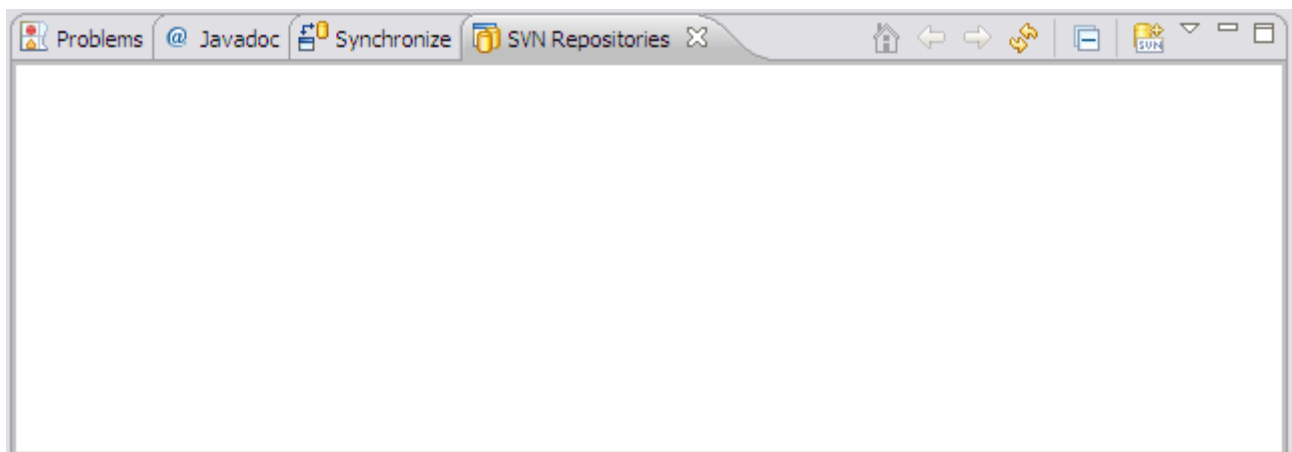


Figure 20 : Ajout de la vue « Svn Repositories »

- Ajouter un nouveau dépôt de source

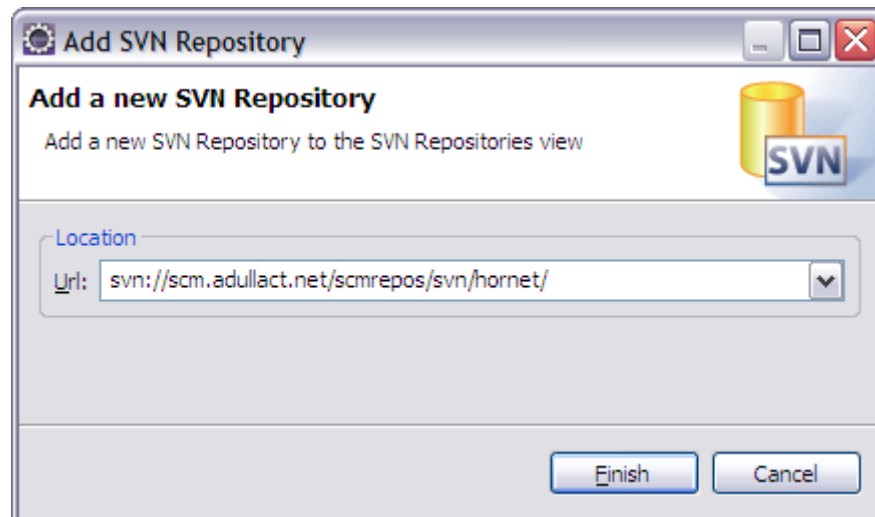


Figure 21 : Sélection du nouveau dépôt

- Récupérer le projet souhaité

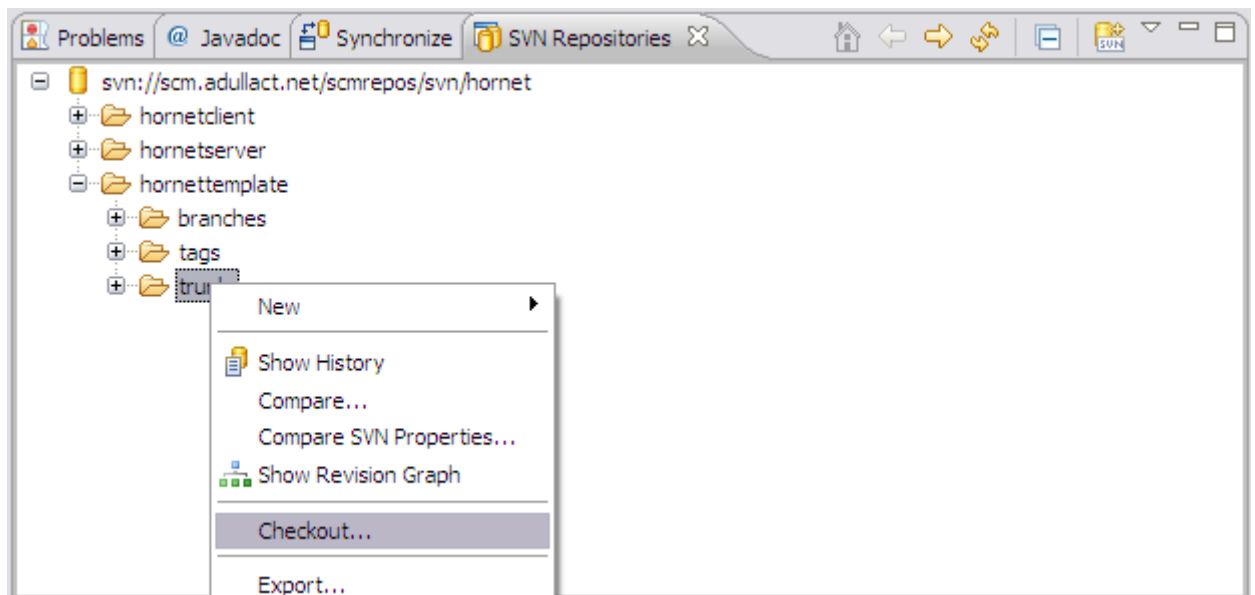


Figure 22 : Récupération locale du nouveau dépôt

3.5 Ajout d'un serveur Tomcat pour le développement

- Afficher la vue **Servers** en sélectionnant dans le menu **Window** → **Show view** → **Other ...** puis **Server** → **Servers** :

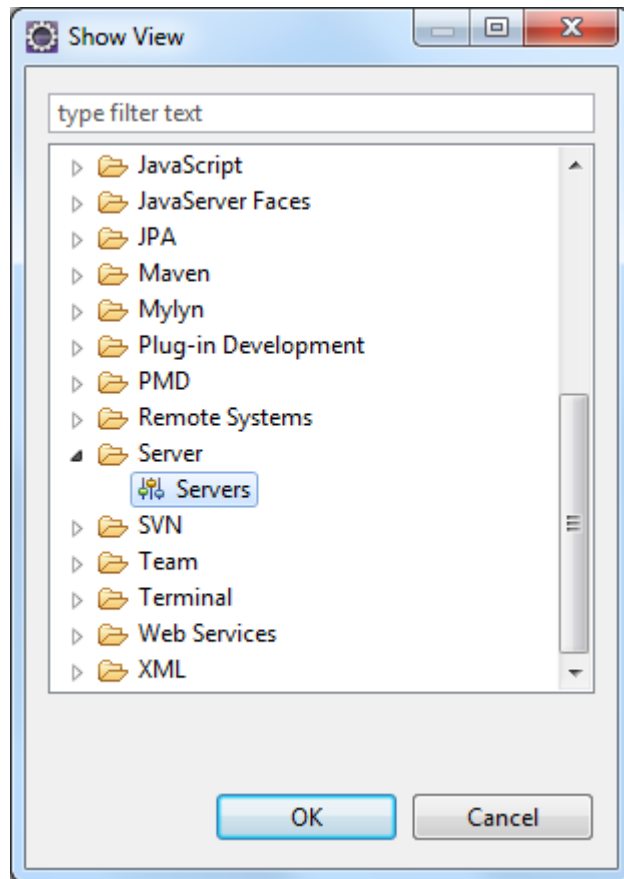


Figure 23 : Afficher la vue "Servers"

- Valider
- Faire un clic droit dans cette nouvelle vue puis choisir **New → Server** :



Figure 24 : Création d'un nouveau serveur de test

- Choisir **Tomcat v6.0 Server**, renseigner un nom et choisir **Apache Tomcat v6.0** dans le champ Server Runtime Environment :

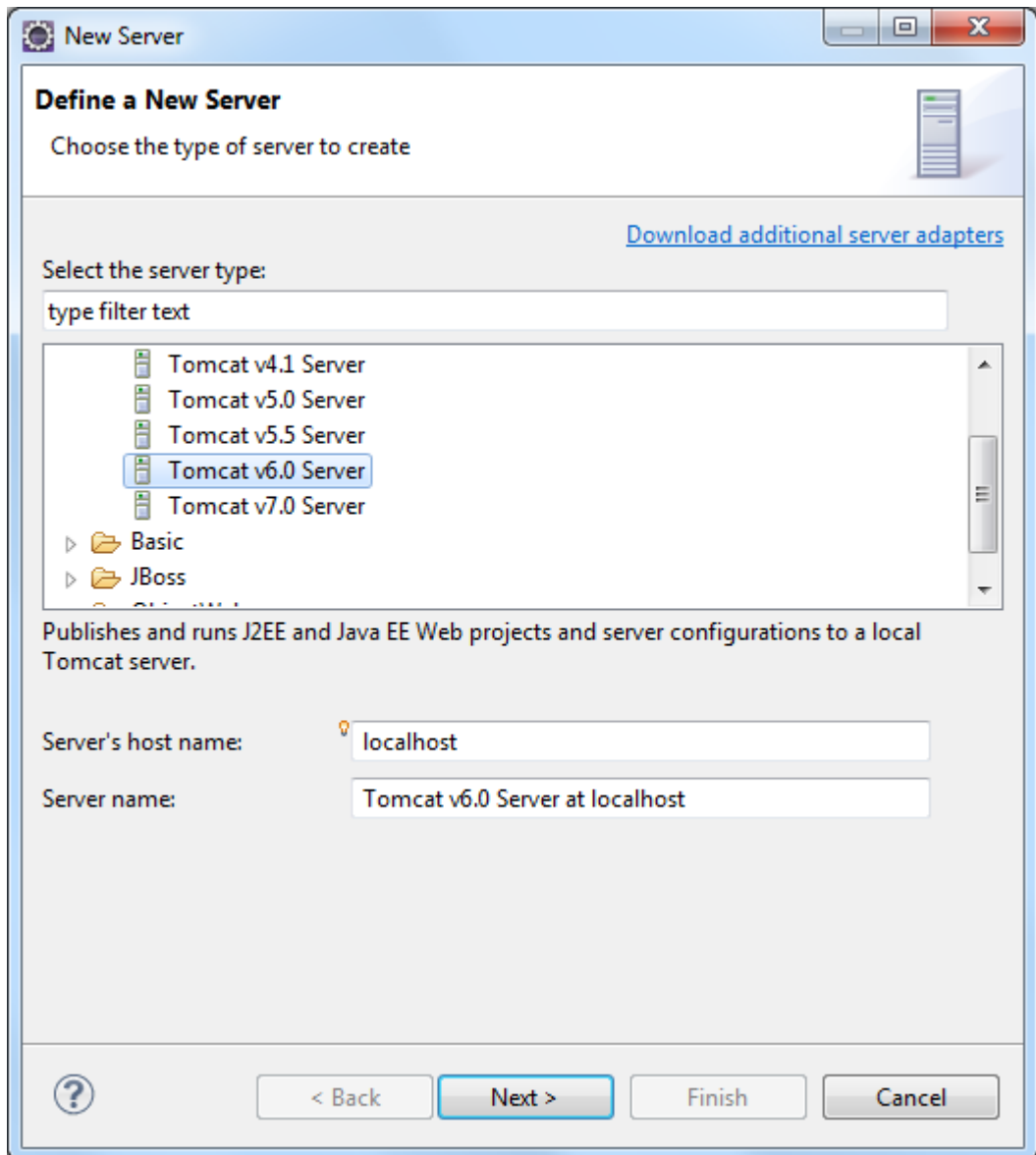


Figure 25 : Paramétrage d'un serveur de test

- Cliquer sur **Next** et choisir les projets à ajouter sur ce serveur de développement :

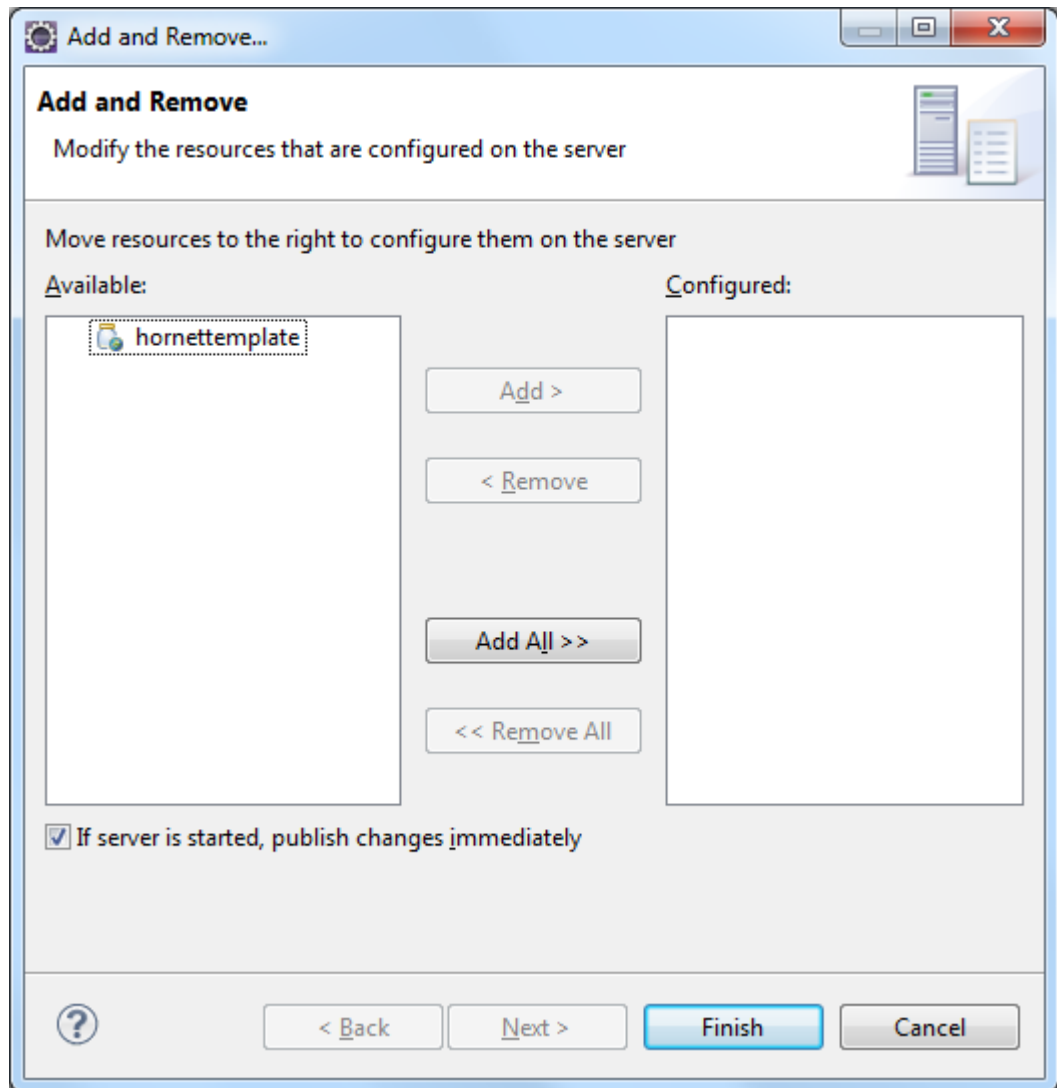


Figure 26 : Choix des applications à ajouter

- Cliquer sur **Finish** le serveur de test créé apparaît alors dans la vue Server:

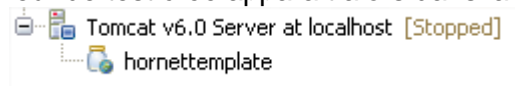


Figure 27 : Serveur de test prêt

Le serveur de test créé peut maintenant être lancé depuis Eclipse.

4 Repositories Ivy

4.1 Présentation d'Ivy

Ivy permet la gestion des dépendances des projets Hornet et s'appuie sur la notion de repository. Un repository entrepose les différents packages (artefacts) des composants qu'il met à disposition ainsi que des fichiers xml Ivy décrivant, entre autres, leurs dépendances.

La configuration se base principalement sur deux fichiers présents dans les projets Hornet :

- `ivysettings.xml` : contient les paramètres indiquant l'emplacement du ou des repository(ies) et les patterns d'url pour les artefacts et les fichiers de dépendances.
- `ivysettings.properties` : Définit principalement le(s) cache(s) à utiliser, les resolvers Ivy (composant chargé de récupérer les dépendances localement ou à distance).

4.2 Installation et configuration

4.2.1 En local

4.2.1.1 Récupération des repositories

Hornet est actuellement fourni avec un ensemble de repositories Ivy complets sous forme d'un fichier zip splitté.

Depuis la page Adullact d'Hornet : <https://adullact.net/projects/hornet/> :

- Choisir la rubrique **Fichiers**
- Dans la section **Repository**, télécharger les deux fichiers **Repository_AAAAMMJJ.zip.XXX** les plus récents (AAAAMMJJ correspondant à la date de création)
- Les dézipper en utilisant 7-Zip dans le répertoire de votre choix.

4.2.1.2 Configuration d'un projet

Dans un projet initialisé à partir du gabarit HornetTemplate, éditer le fichier `ivysettings.properties` et modifier les paramètres au format `repository.XXXX.url` pour indiquer l'emplacement du répertoire racine des repositories.

Pour plus de détail, se référer au Guide de Création d'un Projet.

4.2.2 Depuis un répertoire réseau

Les resolvers Ivy de type « filesystem » supportent également les répertoires distants. Si les repositories ont été installés sur un partage réseau, il faut modifier les paramètres « url » précédemment décrits pour pointer sur le partage concerné.

Sous Windows, par exemple, cela se traduit de la façon suivante :

```
repository.technique.url=\\\\Serveur\\partage\\nomrepository
```

4.2.1 Depuis un gestionnaire d'artefacts

Il est possible de déposer les artefacts présent sur la forge dans un gestionnaire spécifique.

Ivy permet d'utiliser les resolvers de type « url » qui interagiront avec le gestionnaire d'artefacts.

Artifactory est conseillé dans le cadre de Hornet pour son support de ivy.

Exemple de configuration :

ivysettings.xml

```
<resolvers>
  <chain name="main" returnFirst="true" >
```

```
<url name="repository-metier" m2compatible="false" >
  <ivy pattern="${repository.metier.url}/${repository.metier.ivy.pattern}" />
  <artifact pattern="${repository.metier.url}/${repository.metier.artifact.pattern}" />
</url>

<url name="repository-technique" m2compatible="false" >
  <ivy pattern="${repository.technique.url}/${repository.technique.ivy.pattern}" />
  <artifact pattern="${repository.technique.url}/${repository.technique.artifact.pattern}"
/>
</url>

...

</chain>
</resolvers>
```

Ivysettings.properties :

```
repository.host=
repository.url=https://${repository.host}/artifactory

repository.metier.url=${repository.url}/repository-metier
repository.metier.artifact.pattern=${default.artifact.pattern}
repository.metier.ivy.pattern=${default.ivy.pattern}

repository.technique.url=${repository.url}/repository-technique
repository.technique.artifact.pattern=${default.artifact.pattern}
repository.technique.ivy.pattern=${default.ivy.pattern}
```

5 Composants complémentaires

Lors des phases de génération des documentations Java, l'outil Graphviz doit nécessairement être présent dans l'environnement d'exécution afin d'assurer la bonne génération des javadocs.

5.1 Installation de Graphviz

- Télécharger le programme d'installation disponible à l'adresse suivante : <http://www.graphviz.org/pub/graphviz/stable/windows/graphviz-2.30.1.msi>¹
- Démarrer le programme d'installation
- Valider l'écran d'accueil avec le bouton « **Next** »
- Renseigner l'emplacement d'installation du programme

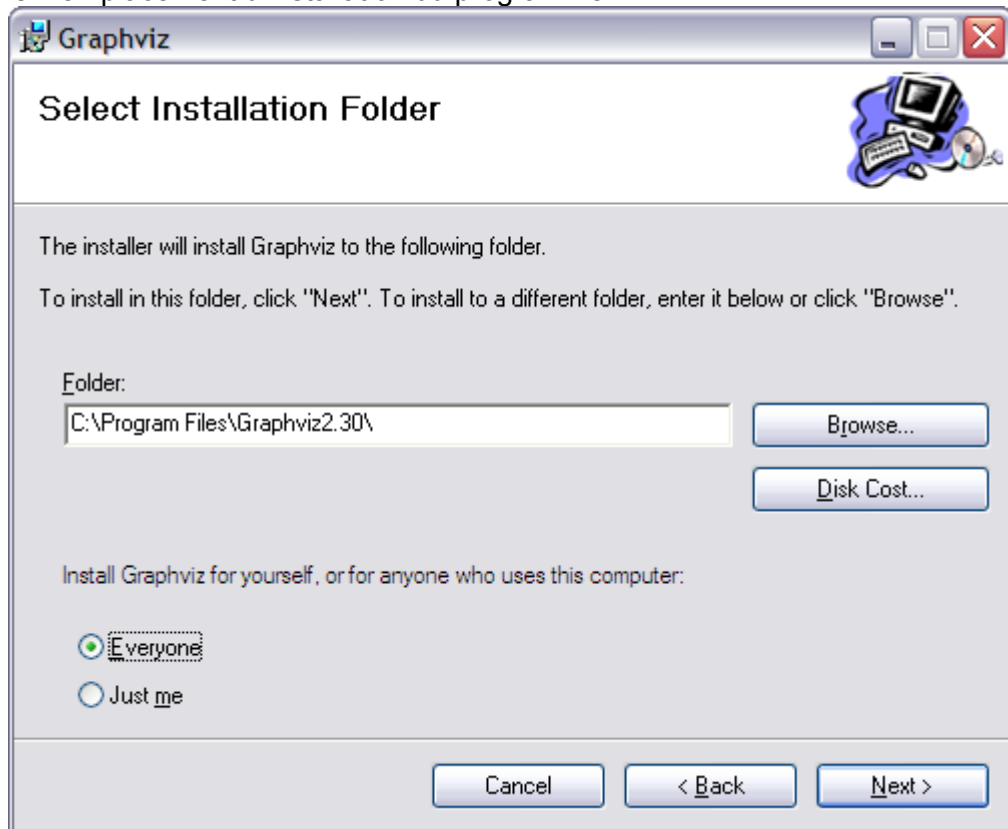


Figure 28 : Installation de Graphviz : Emplacement

- Confirmer l'installation avec le bouton « **Next** »

5.2 Vérification de Graphviz

- Ouvrir une invite de commandes Windows
- Saisir la commande « **dot -V** » et vérifier le bon résultat d'exécution

```
C:\>dot -V
dot - graphviz version 2.30.1 (20130214.1330)
```

¹ Pour les dernières recommandations concernant la procédure d'installation et les particularités liés aux systèmes d'exploitation, se référer à la page : <http://www.graphviz.org/Download..php>

En cas d'erreur, vérifiez que le chemin d'installation de Graphviz est bien présent dans la valeur du champ « **PATH** » des variables d'environnement de Windows.