



TP4 Spring : Programmation Orientée Aspect avec AspectJ / AJDT

Table des matières

Propriété du document.....	3
Prérequis	3
Liens utiles.....	3
Objectifs.....	3
Partie 1 : Installation AJDT.....	4
Introduction à la création d'aspects AspectJ.....	5
Programme principal : HelloWorld.....	5
Aspect : World.....	6
Vue Eclipse : Outline.....	6
Perspective Visualizer.....	6

Propriété du document

Ce document est la propriété de la Société Objis (www.objis.com), spécialisée dans la formation JAVA/J2EE. Il est fourni aux participants des formations SPRING assurée par Objis pour le compte des clients d'Objis.

Prérequis

Les outils et logiciels utilisés se trouvent dans le **répertoire outils** du dossier partagé TSSpring

- JDK 6 (<http://java.sun.com>)
- Eclipse 3.3 europa Entreprise (<http://www.eclipse.org>)
- Spring 2.5 (<http://www.springframework.org/>)
- Spring IDE 2.1.0 (<http://springide.org>)

Pour ce TP Spring IDE doit être au préalable correctement installé.

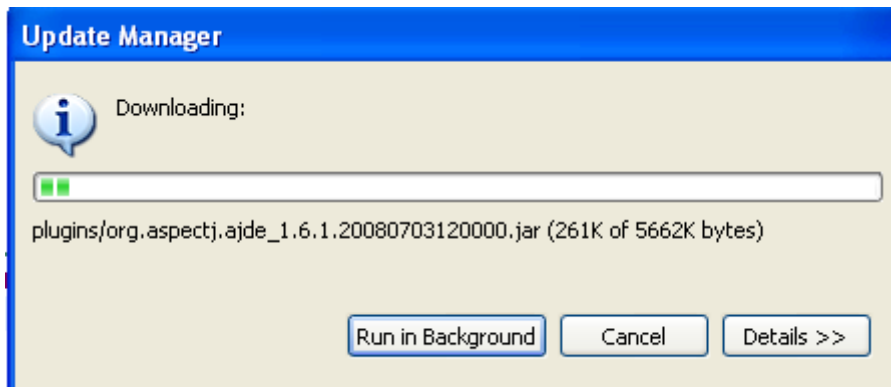
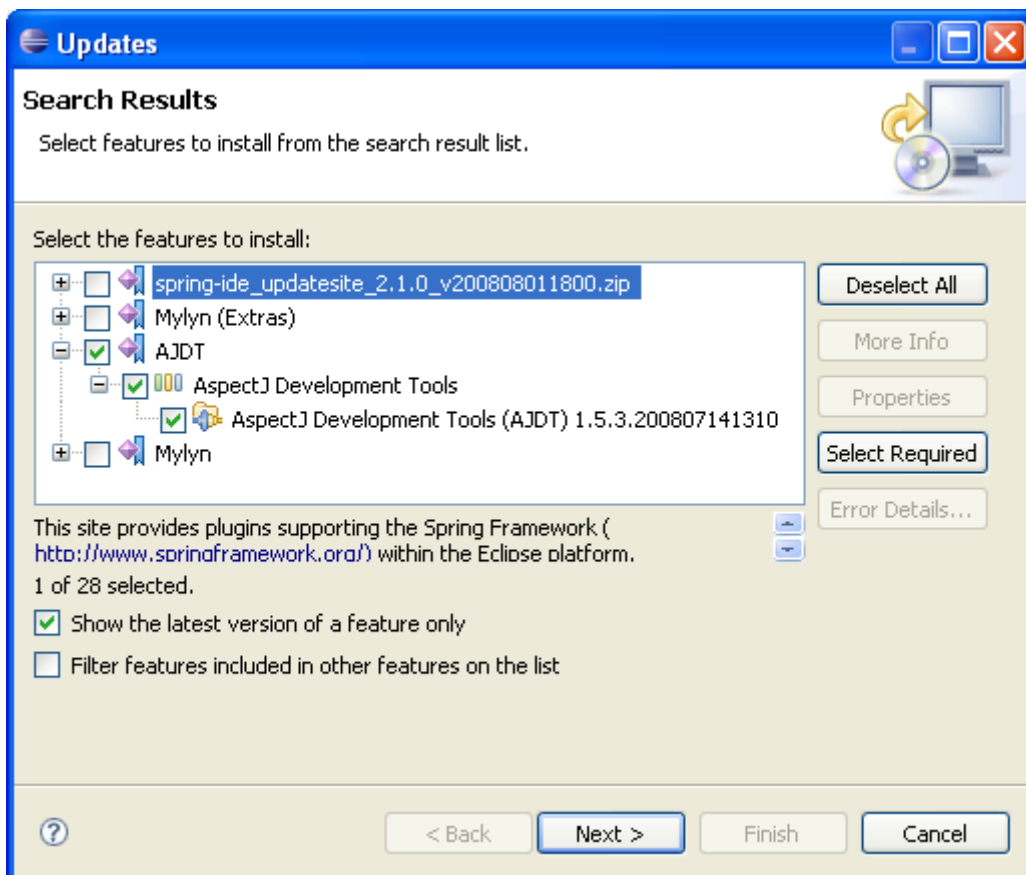
Liens utiles

FAQ outil AJDT : <http://www.eclipse.org/ajdt/faq.php>

Objectifs

- Créer des aspects 100% AspectJ
- Utiliser le Visualiseur d'aspects de AJDT
- Utiliser les vues AJDT
- Debugguer un aspect AspectJ avec AJDT

Partie 1 : Installation AJDT



Partie 2 : Introduction à la création d'aspects avec AJDT

A partir d'eclipse (Menu **Help/Help Contents/AspectJ Developer user guide/Getting started/Basic Tutorial**), faire le premier TP recommandé par AspectJ : Hello AspectJ !

Ce TP va vous permettre de :

- Créer d'un projet Eclipse AspectJ
- Créer un aspect aspectJ
- Voir les préoccupations transversales
- Naviguer entre les points de jonctions
- Lancer un programme avec aspect AspectJ
- Voir graphiquement les aspects à travers la perspective 'Visualisation'

Programme principal : HelloWorld

```
package hello;

public class HelloWorld {

    public void sayHello(){
        System.out.println("Hello");
    }

    public static void main(String[] args) {
        new HelloWorld().sayHello();
    }
}
```

- Expliquez le symbole (couleur orange) présent à gauche de la méthode **sayHello()**
- Que permet un clic droit sur chaque symbole ?



Aspect : World

```
package hello;

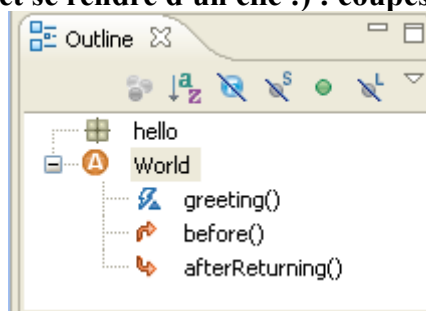
public aspect World {
    pointcut greeting():
        execution(* HelloWorld.sayHello(..));

    before() : greeting() {
        System.out.println("Before advice");
    }
    after() returning : greeting() {
        System.out.println("After advice");
    }
}
```

Vue Eclipse : Outline

Dédiée à un fichier source .aj

On peut voir d'un coup d'oeil (et se rendre d'un clic !) : coupes, greffons



Perspective Visualizer

Menu **Window/Open perspective/other/AspectJ Visualization**

En cliquant sur le projet eclipse, on obtient ceci :



Partie 3 : Développement de préoccupations transversales

- Créez la classe MessageCommunicator, du package com.objjis.aspectj.demo :

```
package com.objjis.aspectj.demo;

public class MessageCommunicator {
    public void deliver(String message) {
        System.out.println(message);
    }

    public void deliver(String person, String message) {
        System.out.println(person + ", " + message);
    }
}
```

Mise en oeuvre service d'Authentification

- Proposez une mise en oeuvre d'un aspect (Ex : SecurityAspect.aj) permettant d'assurer une procédure d'authentification avant toute livraison de message.

Mise en oeuvre service de profiling

- Proposez une mise en oeuvre d'un aspect (Ex : ProfilingAspect.aj) permettant de connaître la durée de livraison d'un message.

Remarque : les informations statiques concernant un point de jonction peuvent être récupérées à partir de l'objet `thisJoinPointStaticPart`

FIN.