

# Guide Pratique EDI NetBeans

Copyright © 2005 Sun Microsystems, Inc. All rights reserved.

## Table des Matières

Création d'un Projet .....	1
Fenêtre Projects.....	3
Configuration du Classpath.....	3
Création d'un Sous-Projet.....	5
Création et Edition de Fichiers.....	6
A Propos de l'Éditeur de Source.....	7
Configuration et Modification de Paquetage Java.....	9
Compilation et Construction.....	9
Voir les Méta-Données du Projet et les Résultats de la "Construction".....	10
Positionnez dans le Source les Erreurs de Compilation.....	11
Exécution.....	11
Création et Execution des Tests.....	12
Déboguer l'Application.....	12
Intégration Commandes Contrôle de Version.....	13
Gestion des Fenêtres de l'EDI.....	15

## Essentiels EDI NetBeans

Ce chapitre fournit un aperçu général du workflow de l'EDI et des parties clés de l'EDI. Une fois que vous aurez terminé ce chapitre, vous devriez avoir une compréhension solide des principes de l'EDI et être capable de tirer profit des fonctionnalités principales de l'EDI.

Si vous êtes déjà familiarisé avec l'EDI NetBeans (4.0 ou plus), vous pouvez probablement survoler ce chapitre ou carrément le sauter. Les chapitres qui suivent vont revenir sur la plupart des points abordés ici, mais plus en profondeur pour répondre aux questions plus complexes, et fournir des détails supplémentaires que vous pouvez utiliser pour arracher plus de productivité de l'EDI.

---

### Création d'un Projet

Avant que vous ne puissiez faire un travail sérieux dans l'EDI, vous devez définir un projet. Le projet définit essentiellement un contexte dans lequel vous pouvez écrire, compiler, tester et

déboguer vos applications. Ce contexte comprend le classpath, les dossiers avec vos sources et tests, et un script Ant avec les “targets” pour la compilation de l'application, l'exécution des tests, la création de fichiers JAR (ou d'autres types de fichiers d'archive distribuables).

Vous pouvez choisir parmi une variété de catégories de modèles de projet, qui sont regroupés selon la technologie sur laquelle vous basés votre application (exemple: standard Java, J2EE Web tier, J2EE Enterprise tier, J2ME).

Dans les catégories de modèles, vous avez des modèles pour de nouvelles applications, et pour instaurer un projet EDI pour des applications existantes sur lesquelles vous travaillez. L'assistant New Project fournit une description pour chaque modèle.

Les modèles “With Existing Sources” de chaque catégorie vous permet d’instaurer des projets EDI standards autour des applications que vous avez développés dans un environnement différent.

Les modèles “With Existing Ant Script” de chaque catégorie vont une étape plus loin et vous permet d’instaurer un projet entièrement basé sur un script Ant existant. Cette approche requiert quelques configuration manuelle pour que certaines fonctionnalités (comme le débogage) fonctionne avec le script Ant. Mais le résultat est que vous avez un EDI qui fonctionne avec tout type de structure de projet, même s'il n'adhère pas aux conventions d'un projet standard de l'EDI.

Pour instaurer un projet:

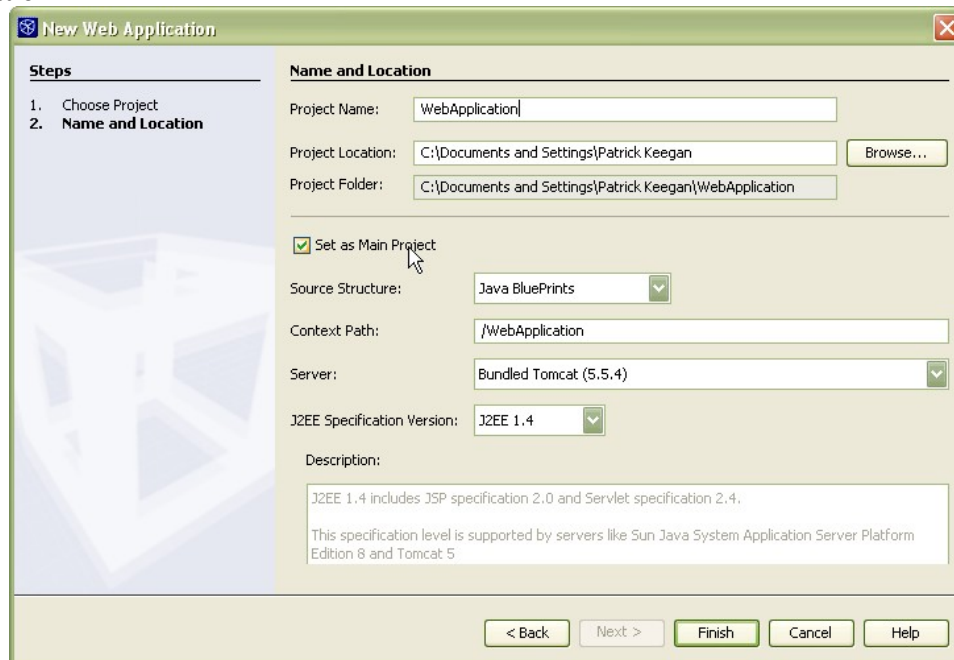
[n] Choisissez File | New Project.

[n] Dans l'assistant, sélectionnez un modèle pour votre projet, et compléter l'assistant.

Les champs qui vous sera demandé de compléter dépendent du modèle.

Typiquement, vous avez besoin de spécifier un emplacement pour le projet (ou, dans le cas d'un projet qui utilise des sources existantes, l'emplacement de ces sources). Les projets Web, Entreprise, et Mobilité incluent également des champs appropriés pour ces types spécifiques d'applications.

## Illustration 2-1



*Assistant New Project, Modèle Web Application, Page Name and Location*

Lorsque vous créez un projet, typiquement l'EDI va faire les choses suivantes pour vous:

- [lb] Créer une arborescence de source avec une classe squelette dedans.
- [lb] Créer un dossier pour les tests unitaires.
- [lb] Créer un script Ant (`build.xml`), qui contient les instructions que l'EDI utilise lorsque vous exécutez des commandes sur votre projet, comme la compilation de fichiers sources, l'exécution de l'application, l'exécution des tests, débogage, génération de la documentation Javadoc, et des fichiers JAR.

Vous pouvez trouver plus d'information quant à la configuration des projets dans le Chapitre 3.

---

## ***Fenêtre Projects***

La fenêtre Projects est essentiellement le centre de commande de votre projet. Il est organisé comme une arborescent de noeuds qui représentent des parties de votre projet. Il fournit un point d'entrée pour vos fichiers ainsi que les options de configuration pour l'application que vous développez.

En plus d'afficher des noeuds pour les fichiers dans l'application que vous développez, il affiche également des noeuds pour les bibliothèques significatives pour votre application. Les noeuds de bibliothèque montrent la version de JDK pour laquelle vous développez ainsi que les autres bibliothèques sur lesquelles votre projet se base.

La fenêtre Projects vous présente votre projet sous une forme "logique". C'est à dire qu'il représente les éléments de votre application conceptuellement (plutôt que littéralement). Par exemple, les sources Java sont regroupés en packages, sans noeuds pour chaque niveau de hiérarchie de fichier. Les fichiers que vous ne devez normalement pas voir, comme les classes Java compilées et les fichiers de méta-données du projet sont cachés. Cela simplifie l'accès aux fichiers sur lesquels vous travaillez régulièrement. De plus, la fenêtre Projects fournit un noeud Libraries, qui vous donne une vue de votre classpath.

Si vous désirez naviguer dans la structure physique du projet, y compris les méta-données du projet, les classes compilées, les fichiers Jar et d'autres fichiers créés lors du build, ouvrez la fenêtre Files.

---

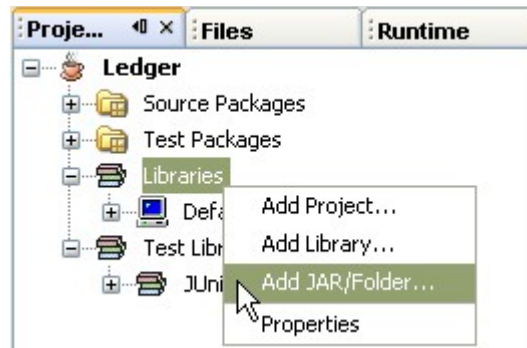
## ***Configuration du Classpath***

Lorsque vous créez un projet, l'EDI définit un classpath par défaut pour vous, basé sur le modèle de projet que vous avez sélectionné. Si vous avez d'autres choses à ajouter à votre classpath, vous pouvez le faire via le noeud Libraries du projet.

En fait, l'EDI fait la distinction entre plusieurs types de classpath, selon le type de projet, comme le classpath de compilation, le classpath de compilation des tests, le classpath d'exécution, et le classpath d'exécution des tests. Le classpath de compilation sert typiquement de base pour les autres classpaths (c'est à dire que les autres classpaths héritent de ce qui a dans le classpath de compilation).

Pour ajouter un élément au classpath de compilation (et donc également aux autres classpaths), cliquez-droit sur le noeud Libraries du projet et choisissez Add JAR/Folder.

**Illustration 2-2**



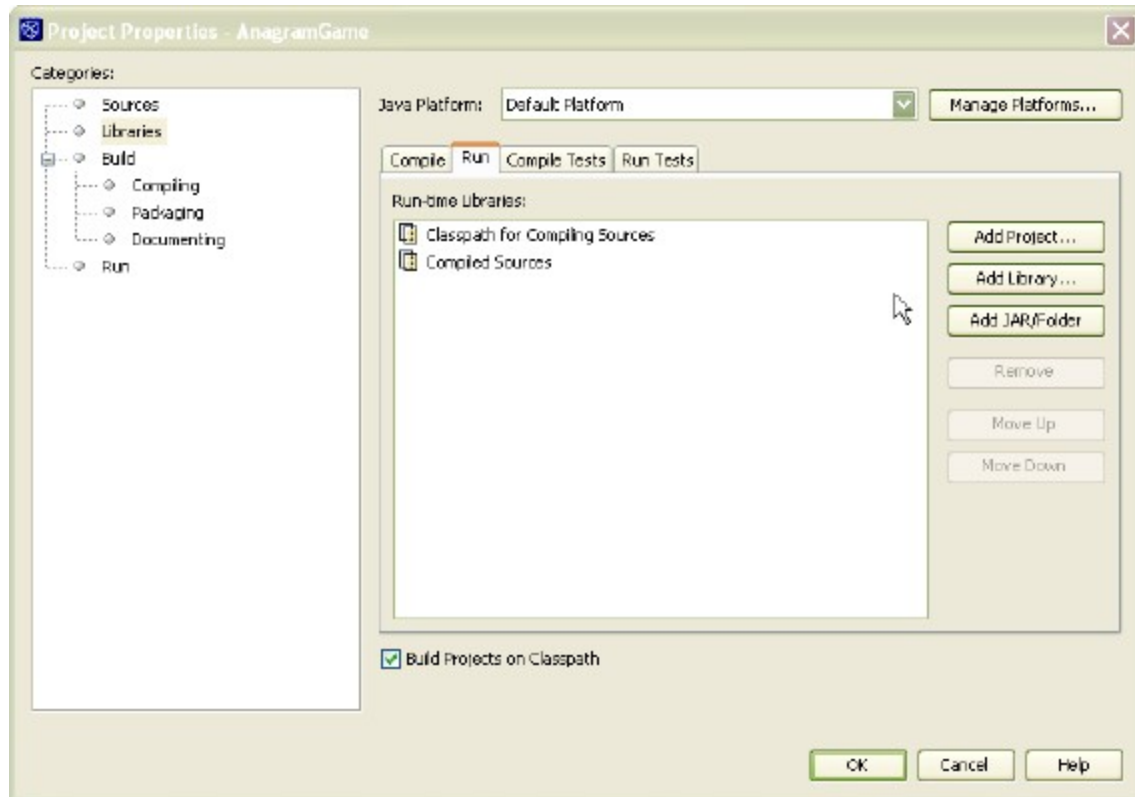
*Fenêtre Projects. Ajout d'un fichier JAR au classpath.*

### **Conseil EDI NetBeans**

Lorsque vous cliquez-droit sur le noeud Libraries, vous pouvez également rajouter Add Project ou Add Library. Lorsque vous rajoutez un projet, vous ajoutez le résultat du projet (comme un fichier JAR) au classpath.

You can edit other classpaths in the Properties dialog box for a project. To open the Project Properties dialog box, right-click the project's node in the Projects window and choose Properties. In the dialog box, click the Libraries node and use the customizer in the right panel to specify the different classpaths.

Si vous choisissez Add Library, vous pouvez rajouter une des "libraries" reconnues par le gestionnaire de bibliothèques de l'EDI. Dans ce contexte, les bibliothèques sont essentiellement juste un moyen de regrouper un ou plusieurs fichiers JAR, leurs sources et/ou leur documentation Javadoc. Vous pouvez gérer des bibliothèques existantes et en concevoir de nouvelles dans le Gestionnaire de Bibliothèques, que vous pouvez ouvrir en choisissant Tools | Library Manager.



**Illustration 2-3**  
*Boîte de Dialogue Project Properties, page Libraries.*

---


## Création d'un Sous-Projet

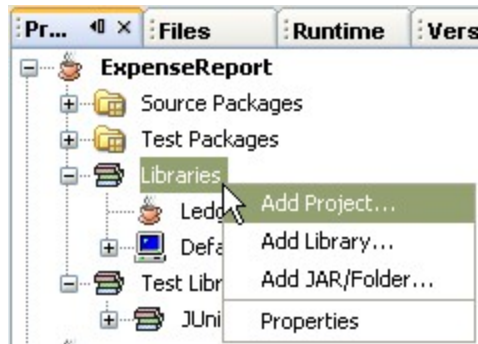
Bien qu'il n'y ait pas de distinction explicite dans l'EDI entre un projet et un “sous-projet”, vous pouvez créer une hiérarchie de projets en spécifiant des dépendances entre les projets. Par exemple, vous pouvez créer un projet Application Web ombrelle qui chapeaute un ou plusieurs projet Java Class Library. Pour des applications plus larges, vous pourriez avoir plusieurs niveaux de dépendance de projet.

Pour définir des dépendances entre les projets:

[nl] Cliquez-droit sur le noeud Libraries du projet et choisissez Add Project.

Once you have established this dependency, the distributed outputs (such as JAR files) of the “added” project become part of the other project's classpath.

[nl] Dans la boîte de dialogue qui apparaît, naviguez jusqu'au dossier du projet dont vous désirez dépendre. Les dossiers du projet sont désignés par l'icône .



#### **Illustration 2-4**

*Fenêtre Projects. Crée une dépendance d'un projet sur un autre.*

#### **Conseil EDI NetBeans**

Il n'y a pas de distinction visuelle entre projet et sous projets dans l'EDI, mais il y a un concept de projet "principal". Le projet principal dans l'EDI est simplement celui que l'EDI traite comme le point d'entrée pour les commandes principales comme Build Main Project and Run Main Project. Le projet principal est indiqué avec une police en gras dans la fenêtre Projects.

Il ne peut y avoir qu'un seul projet principal à la fois, bien qu'il soit possible d'avoir de multiples projets ouverts en même temps (y compris des projets ombrelles qui servent de points d'entrée pour les autres applications que vous développez).

Vous pouvez faire d'un projet le projet principal en cliquant-droit sur son noeud dans la fenêtre Projects et en choisissant Set Main Project.

---

## ***Création et Edition de Fichiers***

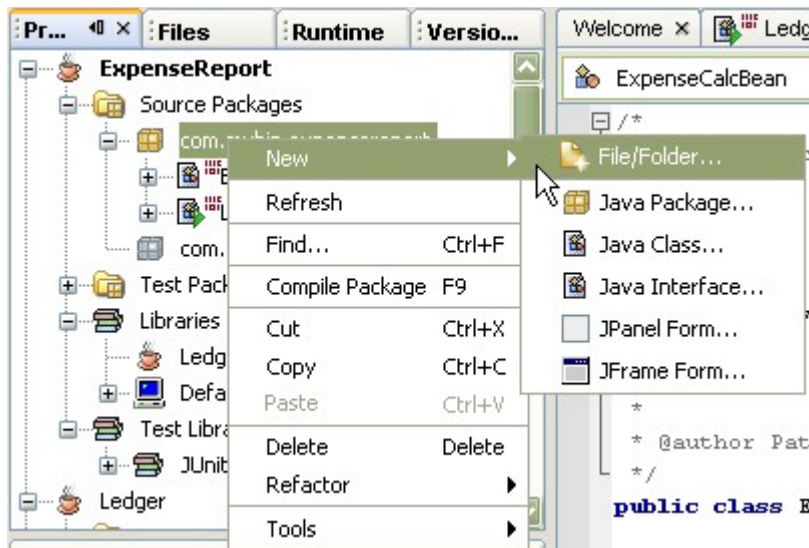
Une fois que vous avez défini un projet, vous pouvez ajouter des fichiers à votre projet et commencer son édition. Vous pouvez ajouter des fichiers à un projet en les créant depuis l'assistant Nouveau Fichier.

Pour ouvrir l'assistant Nouveau Fichier, faites l'une des choses suivantes:

- [lb] Dans la fenêtre Projects, cliquez-droit sur le noeud Source Packages (ou l'un des noeuds de paquetage qu'il comprend) et choisissez un des modèles depuis le sous-menu New. Si aucun des modèles présentés ne vous convient, choisissez File/Folder (comme montré dans l'illustration 2-5) pour ouvrir un assistant avec une sélection complète des modèles disponibles.
- [lb] Choisissez File | New File pour ouvrir l'assistant New File.

#### **Illustration 2-5**

*Fenêtre Projects. Création d'un nouveau fichier*



Dans l'assistant New File, vous pouvez nommer le fichier et spécifier un dossier. Pour les classes Java, vous pouvez désigner un nom de paquetage comprenant des points (au lieu des barres obliques délimitant les noms des dossiers).

Une fois que vous avez fini avec l'assistant, le fichier s'ouvre dans un onglet de l'aire d'édition, à droite de la fenêtre Projects.

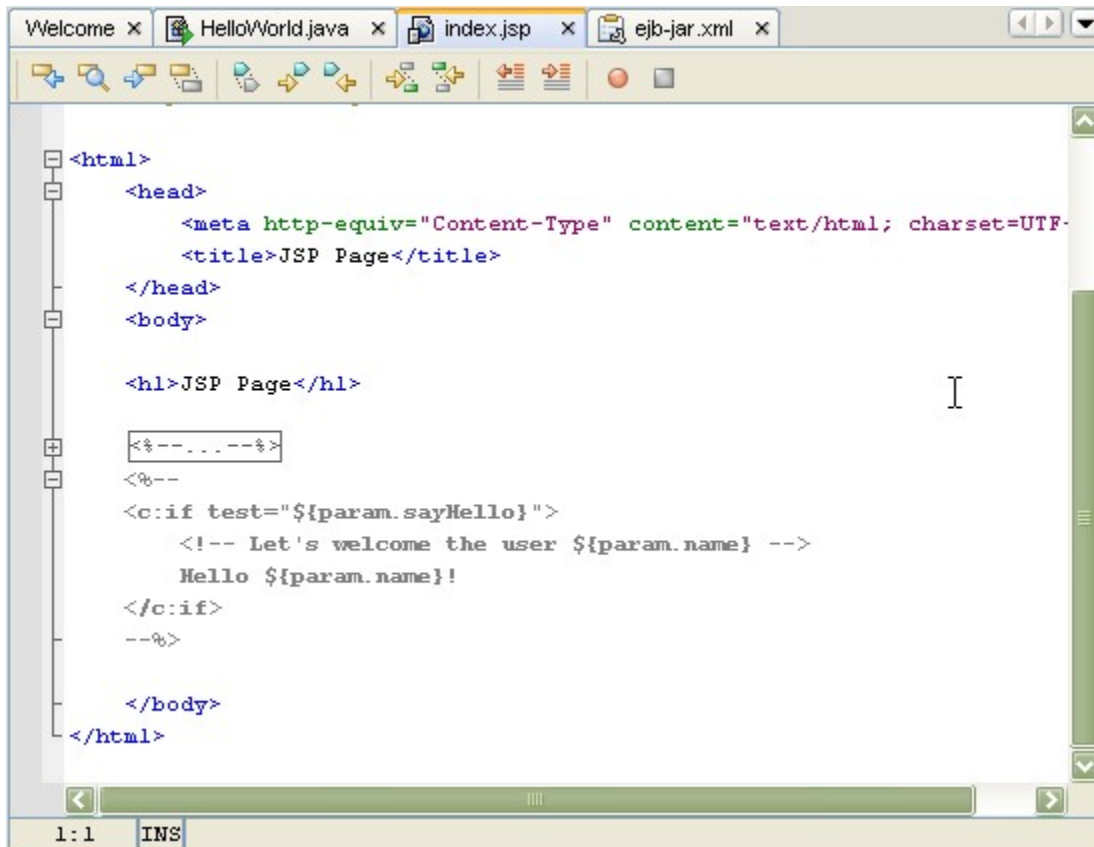
---

## ***A Propos de l'Éditeur de Source***

L'Editeur de Source est le point central de l'EDI, là où vous allez écrire et générer du code. L'Éditeur de Source est en fait une collection de différents types d'éditeurs avec différents buts. Il y a des éditeurs de texte pour différents types de fichiers, comme les fichiers Java, JSP (comme montré sur l'illustration 2-6), XML, HTML, et les fichiers textes.

### **Illustration 2-6**

*Fenêtre Editeur de Source avec fichier JSP ouvert.*



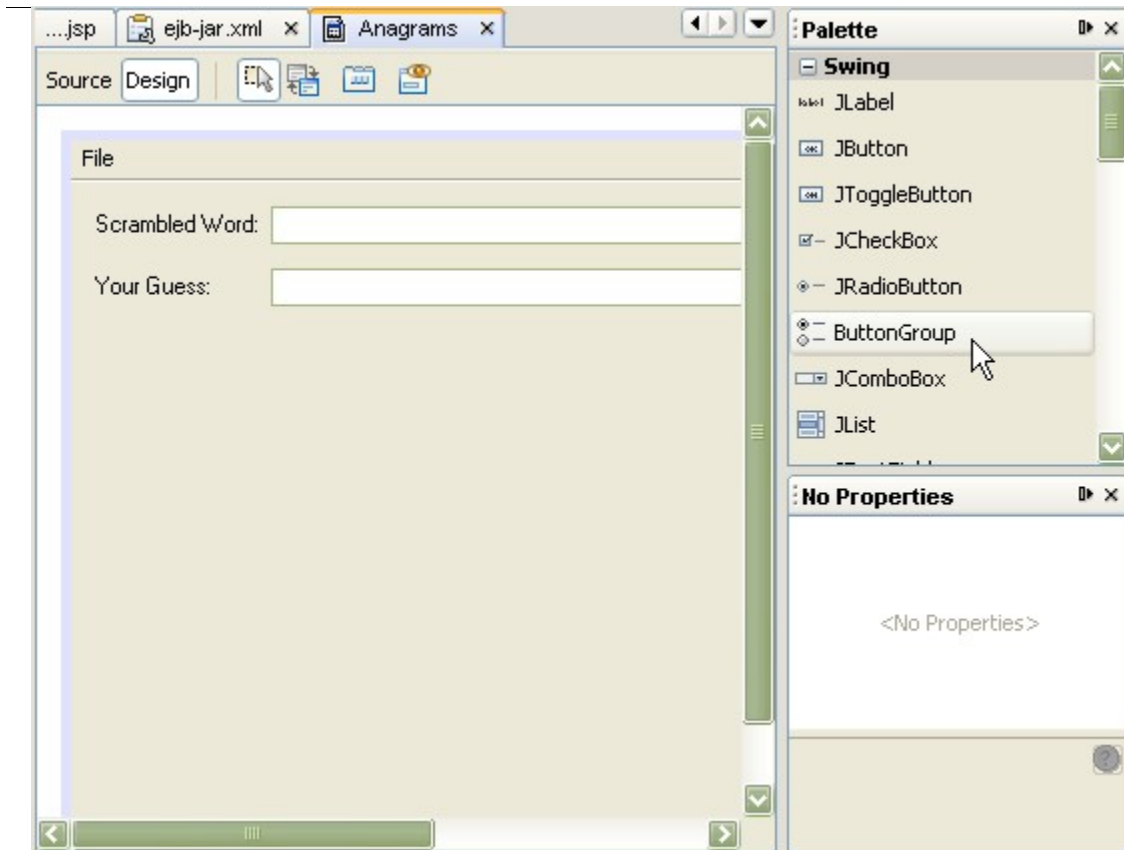
Il y a également des éditeurs visuels pour les forms AWT et Swing, les descripteurs de déploiement, et autres types de fichiers, bien qu'il est également possible d'éditer directement le source de ces types de fichier.

Par exemple, les modèles GUI comme JPanel Form et JFrame Form s'ouvre dans un éditeur visuel (comme montré dans l'illustration 2-7) avec les fenêtres Palette, Inspector, et Propriétés. Vous pouvez cliquer sur le bouton Source dans la barre d'outil pour accéder au source du fichier.

#### **Illustration 2-7**

*Editeur Visuel de Form dans la fenêtre d'Éditeur de Source.*





## **Configuration et Modification de Paquetage Java**

Vous pouvez configurer un paquetage Java grâce aux assistants New Project et New File. Vous pouvez également créer des paquetages sans utiliser ces assistants.

Pour créer un nouveau paquetage, cliquez-droit sur le noeud Source Packages dans votre projet et choisissez New | Java Package. Dans l'assistant, remplissez un nom de paquetage, délimité par un point(exemple: `com.masociete.monapplication`).

Vous pouvez ensuite déplacer des classes dans ce paquetage via couper/coller ou en déplaçant leurs noeuds à la souris.

### **Conseil EDI NetBeans**

Lorsque vous déplacez des classes, la boîte de dialogue Refactor Code for Moved Class s'ouvre et propose de mettre à jour le restant du code dans le projet pour refléter le changement d'emplacement de la classe. Cliquez sur Next pour obtenir un aperçu des modifications dans la fenêtre Refactoring. Cliquez ensuite sur Do Refactoring pour effectuer les changements proposés.

---

## **Compilation et Construction**

Lorsque vous avez défini un projet, l'EDI lui fournit des paramètres par défaut de classpath et de compilation, pour que le projet soit prêt à compiler dès que vous avez ajouté quelques classes au projet.

Vous pouvez compiler un fichier ou paquetage isolément en cliquant droit sur son noeud et en choisissant Compile. Mais généralement, vous allez “construire” le projet dans son entièreté. Selon le type de projet, sa “construction” consiste typiquement à compiler les projets et sous-projets et créer les résultats comme des fichiers JAR pour chacun de ces projets.

Pour “construire” votre projet, cliquez-droit sur le noeud du projet dans la fenêtre Projects et choisissez Build Project. Si ce projet est désigné comme projet principal (le nom du projet est en gras dans la fenêtre Projects), vous pouvez choisir Build | Build Main Project ou pressez F11. Si vous désirez effacer les produits des “builds” précédents avant de reconstruire, choisissez Build | Clean et Build Main Project ou pressez Shift-F11.

Lorsque vous initiez un “build”, l'EDI indique la progression de la construction dans la fenêtre Output sous la forme d'un output Ant.

#### **Conseil EDI NetBeans**

Vous pouvez spécifier les options de compilation dans la boîte de dialogue Project Properties. Cliquez-droit sur le noeud du projet dans la fenêtre Projects et choisissez Properties. Ensuite, cliquez sur le noeud Compiling pour y introduire les options.

---

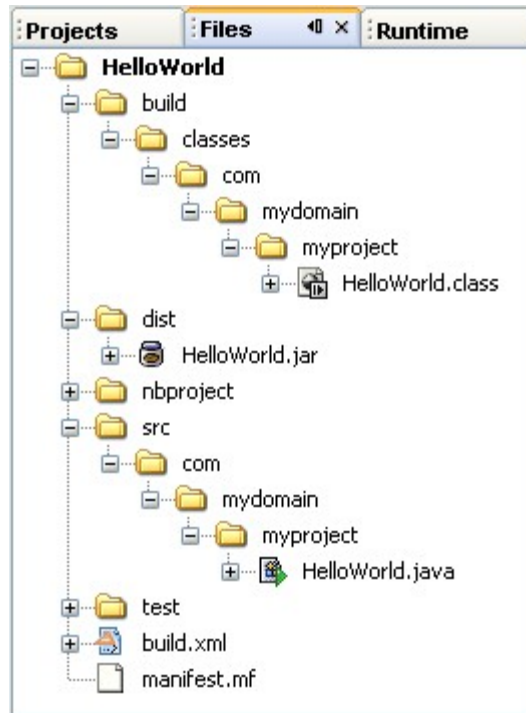
### ***Voir les Méta-Données du Projet et les Résultats de la “Construction”***

Dans la fenêtre Files, vous pouvez voir la structure physique de votre projet, y compris les fichiers de classes compilées, les fichiers JAR, vos scripts Ant, et autres méta-données du projet.

Les commandes liées au projet (comme Build Project) ne sont pas disponibles depuis les noeuds dans la fenêtre Files, mais d'autres commandes de type “Explorer” comme Open, Cut, et Paste sont présentes.

La fenêtre Files est très utile si vous désirez personnaliser le script pour votre projet ou si vous désirez naviguer parmi les résultats de votre projet. Vous pouvez également examiner le contenu des fichiers JAR créés par votre projet.

L'illustration 2-8 montre la structure de l'application HelloWorld créée au Chapitre 1.



**Illustration 2-8**  
Fenêtre Files. Vue “physique” du projet HelloWorld

## Positionnez dans le Source les Erreurs de Compilation

Si des erreurs de compilation sont reportées lorsque vous compilez, vous pouvez placer le curseur directement sur la ligne provoquant l'erreur de compilation en double-cliquant sur l'erreur apparaissant dans la fenêtre Output sous la forme d'un hyper-lien (comme montré dans l'illustration 2-9) ou en pressant F12.

Si vous avez plusieurs erreurs, vous pouvez utiliser F12 (Erreur Suivante) et Maj-F12 (Erreur Précédente) pour vous déplacer d'une erreur à l'autre.

### Illustration 2-9

```

Output - AnagramGame.jar
-----
deps:jar:
inst:
deps:jar:
compile:
jar:
Compiling 1 source file to C:\Documents and Settings\Fabrick Meegan\AnagramGame\build\classes
E:\Documents and Settings\Fabrick Meegan\AnagramGame\src\com\tor\anagram\ui\Anagram.java:30: cannot find symbol
symbol : variable guessWord
location: class com.tor.anagram.ui.Anagram
    guessWord.requestFocusInWindow();
1 error
BUILD FAILED (total time: 0 seconds)

```

Fenêtre Output montrant une erreur de compilation.

## Exécution

Vous pouvez exécuter l'application que vous développez depuis l'EDI en cliquant-droit sur le noeud du projet et en choisissant Run Project ou en pressant F6.

Vous pouvez exécuter un fichier individuellement en cliquant-droit sur le fichier dans l'Éditeur de Source ou le noeud du fichier dans la fenêtre Projects et choisissant Run File ou en pressant Maj-F6.

Vous pouvez arrêter une application qui s'exécute en ouvrant la fenêtre Runtime, étendant le noeud Processes, et cliquant droit sur le noeud du processus qui s'exécute pour finalement y choisir Terminate Process.

Si vous avez besoin de spécifier une classe principale pour le projet, ou si vous désirez exécuter le projet avec quelques arguments, vous pouvez les spécifier dans la boîte de dialogue Project Properties.

To for the project to run in the IDE. You can do so by right-clicking the project's node in the Projects window, choosing Properties, selecting the Run node, and entering

---

## Création et Execution des Tests

Les modèles de projet de l'EDI sont conçu avec les tests unitaires à l'esprit. La plupart des types de projets crée un dossier, à coté de celui contenant vos sources, pour les test unitaires. Vous pouvez demander à l'EDI de générer le squelette du code pour le test unitaire d'une classe et le placer dans le dossier de test avec une structure de paquetage correspondant à celle de la classe à tester.

Pour générer le code de test unitaire pour une classe:

[nl] Dans la fenêtre Projects, cliquez-droit sur la classe pour laquelle vous désirez créer un test et choisissez Tools | JUnit Tests | Create Tests.

[nl] Dans la boîte de dialogue Create Tests, indiquez un nom de classe et son emplacement, et spécifiez les options de génération de code pour le test.

Par défaut, le nom de la classe est remplis pour vous et correspond au nom de la classe à tester avec `Test` suffixé au nom. Les classes de test sont placées dans un dossier test qui a la même structure de paquetage que vos sources.

Pour exécuter les tests du projet sélectionnés, pressez Alt-F6 ou choisissez Run | Test "*NomDuProjet*".

Pour exécuter un test pour un fichier spécifique, sélectionnez le fichier dans l'Éditeur de Source ou la fenêtre Projects et pressez Ctrl-F6 ou choisissez Run | Run File | Test "*nomDuFichier*".

---

## Déboguer l'Application


Le débogueur de l'EDI vous permet de suspendre l'exécution de votre programme à des points stratégiques ("points d'arrêt") et de vérifiez les valeurs de variables, le status des threads, etc. Une fois que vous avez suspendu l'exécution à un point d'arrêt, vous pouvez avancer pas à pas dans le code.

Pour démarrer le débogage d'un programme:

[nl] Assurez-vous que le programme que vous désirez déboguer est bien défini comme projet principal de l'EDI.

Le nom du projet principal est indiqué en gras dans la fenêtre Projects. Vous pouvez indiquer qu'un projet normal devienne le projet principal en cliquant-droit sur son noeud et en choisissant Set Main Project.

[nl] Déterminez le point dans votre code où vous désirez commencer à déboguer et définissez un point d'arrêt sur cette ligne en cliquant dans la marge de gauche de cette ligne.

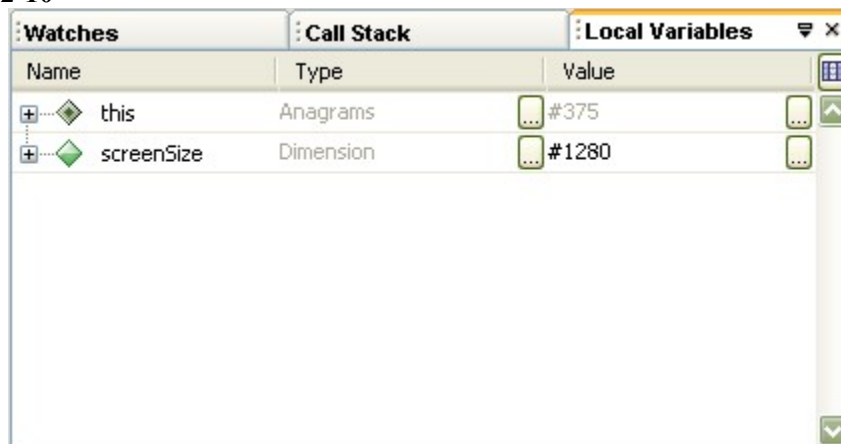
L'icône  apparaît dans la marge de gauche pour marquer le point d'arrêt. De plus, toute la ligne apparaîtra en rose.

[nl] Lancez le débogueur en choisissant Debug Main Project ou en pressant F5.

L'EDI construit (ou reconstruit) l'application et ensuite ouvre la console de débogage en bas de l'EDI et les fenêtres Watches, Call Stack, et Local Variables windows à sa droite.

[nl] Cliquez sur la fenêtre Local Variables (illustrée sur l'illustration 2-10) pour voir la valeur de n'importe quelle variable actuellement dans le scope du programme.

### Illustration 2-10



Fenêtres du Débogueur, avec la fenêtre Local Variables ayant le focus.

---

## Intégration Commandes Contrôle de Version

Si vous avez déjà utilisé un système de version de contrôle pour vos sources, vous pouvez facilement intégrer ces commandes du système dans le workflow de l'EDI. L'EDI fournit le support pour travailler avec de nombreux systèmes de contrôle de version.

L'EDI agit comme un interface graphique pour l'application cliente de contrôle de version que vous utilisez déjà. Lorsque vous appelez les commandes de version de contrôle depuis l'EDI, l'EDI passe ces commandes au client du contrôle de version, qui s'occupe des commandes. L'EDI affiche également tout résultat généré par le client de contrôle de version.

Dans l'EDI NetBeans 4.1, la définition de l'EDI pour travailler avec les sources versionnées est séparée de la définition du projet. Si vous avez déjà des sources provenant d'un système de contrôle de version et que vous désirez que les commandes de contrôle de version soient disponibles dans l'EDI pour un projet, vous devez faire connaître le répertoire de travail versionnée à l'EDI.

### Conseil EDI NetBeans

If you are using a post-4.1 version of the IDE, this process might be streamlined, so that the registration of the version control system in the IDE is coupled with the creation of the project.

Pour indiquer à l'EDI de travailler avec votre système de contrôle de version:

- [nl] Choisissez Versioning | Versioning Manager.
- [nl] Dans la boîte de dialogue Versioning Manager, cliquez sur Add.
- [nl] Sélectionnez le système de contrôle de version que vous utilisez depuis la liste déroulante Profile et indiquez l'emplacement du répertoire de travail.  
Si vous avez plusieurs projets dans le même répertoire de travail, vous pouvez sélectionner le répertoire racine pour faire connaître le contrôle de version pour tous les projets en une seule opération.
- [nl] Vérifiez les paramètres du serveur remplis par l'EDI et ajoutez les paramètres manquants.  
Si vous utilisez CVS comme système de contrôle de version, vous avez l'option d'utiliser un client intégré plutôt qu'un exécutable CVS séparé.
- [nl] Cliquez sur Finish pour quitter l'assistant et cliquez ensuite sur Close pour quitter le Gestionnaire de Version.
- [nl] Si vous ne l'avez pas encore fait, créez un projet dans l'EDI pour vos sources via l'assistant New Project pour que vous puissiez développer ces sources dans l'EDI.

#### **Conseil EDI NetBeans**

Si aucun profile n'est disponible dans l'assistant pour le système de contrôle de version que vous utilisez, vous pourriez trouver un profile en-ligne à l'adresse <http://vcsgeneric.netbeans.org/profiles/index.html>.

Vous pouvez également créer votre propre profile en choisissant le profile Empty dans le Gestionnaire de Version en ensuite le personnaliser pour qu'il fonctionne avec votre système de contrôle de version. Voyez la page <http://vcsgeneric.netbeans.org/doc/profiles/index.html> pour des informations sur la création de profile pour votre système de contrôle de version.

Voir Chapitre 3: Configurer un Projet pour Travailler avec Contrôle de Version pour plus d'informations sur le contrôle de version avec l'EDI, y compris les informations sur comment mettre vos méta-données de projet sous contrôle de version.


Une fois que vous avez configuré un répertoire de travail sous un contrôle de version de cette manière, la fenêtre Versioning apparaît à l'endroit occupé par la fenêtre Projects. Vous pouvez exécuter des commandes de contrôle de version depuis cette fenêtre. Cependant, vous ne pouvez pas exécuter des commandes liées au projet ou faire des choses de type "Explorer" avec les fichiers comme Ouvrir, Copier, ou Coller.

Si vous avez déjà défini un projet EDI pour ces sources, un sous-menu avec les commandes de contrôle de version apparaîtra dans le menu contextuel (disponible via un clic droit) de tous les noeuds du projet dans la fenêtre Projects.

---

## *Gestion des Fenêtres de l'EDI*

Le système de fenêtrage de l'EDI est conçu pour fournir un layout cohérent et non intrusif des nombreuses fenêtres dont vous avez besoin tout en vous permettant d'ajuster le layout sans trop d'effort selon vos besoins. Voici certaines choses que vous pouvez faire:

- [lb] Redimensionner les fenêtres en cliquant sur un bord de la fenêtre et la tirer jusqu'à ce qu'elle atteigne la taille que vous désirez.
- [lb] Agrandir une fenêtre dans l'EDI en double-cliquant sur son onglet (vous pouvez revenir à son apparence précédente en double-cliquant à nouveau sur l'onglet). Vous trouverez cette fonctionnalité particulièrement intéressante dans l'Éditeur de Source.
- [lb] Déplacer une fenêtre vers une autre partie de l'EDI en cliquant sur son onglet et en la lâchant dans un autre coin de l'EDI.
- [lb] Utiliser Tirer/Lâcher pour découper une fenêtre en deux.
- [lb] Rendre une fenêtre "coulissante" en cliquant sur son bouton . Lorsque vous cliquez sur son bouton, la fenêtre est minimisée avec un bouton représentant cette fenêtre placée sur l'un des bords de l'EDI. Vous pouvez mettre le curseur de la souris sur le bouton pour afficher temporairement la fenêtre, ou vous pouvez cliquer sur le bouton pour ouvrir la fenêtre.