# Prise en main d'Ant

## Table des matièresPlier

* [I. Introduction](https://skebir.developpez.com/tutoriels/java/ant/#LI)
* [II. Création du projet par ligne de commande](https://skebir.developpez.com/tutoriels/java/ant/#LII)
  + [II-A. Compilation](https://skebir.developpez.com/tutoriels/java/ant/#LII-A)
  + [II-B. Création du fichier jar exécutable](https://skebir.developpez.com/tutoriels/java/ant/#LII-B)
  + [II-C. Exécution du projet](https://skebir.developpez.com/tutoriels/java/ant/#LII-C)
* [III. Création du projet avec Ant](https://skebir.developpez.com/tutoriels/java/ant/#LIII)
  + [III-A. La cible clean](https://skebir.developpez.com/tutoriels/java/ant/#LIII-A)
  + [III-B. La cible compile](https://skebir.developpez.com/tutoriels/java/ant/#LIII-B)
  + [III-C. La cible jar](https://skebir.developpez.com/tutoriels/java/ant/#LIII-C)
  + [III-D. Exécution du projet](https://skebir.developpez.com/tutoriels/java/ant/#LIII-D)
  + [III-E. Dépendances entre cibles](https://skebir.developpez.com/tutoriels/java/ant/#LIII-E)
  + [III-F. Récapitulatif](https://skebir.developpez.com/tutoriels/java/ant/#LIII-F)
  + [III-G. Amélioration du fichier build.xml](https://skebir.developpez.com/tutoriels/java/ant/#LIII-G)
* [IV. Comparaison entre build.xml et la ligne de commande](https://skebir.developpez.com/tutoriels/java/ant/#LIV)
* [V. Autres outils](https://skebir.developpez.com/tutoriels/java/ant/#LV)
  + [V-A. Maven](https://skebir.developpez.com/tutoriels/java/ant/#LV-A)
* [VI. Liens utiles](https://skebir.developpez.com/tutoriels/java/ant/#LVI)
* [VII. Téléchargement de sources](https://skebir.developpez.com/tutoriels/java/ant/#LVII)
* [VIII. Conclusion](https://skebir.developpez.com/tutoriels/java/ant/#LVIII)
* [IX. Remerciements](https://skebir.developpez.com/tutoriels/java/ant/#LIX)

Le but principal de cet article est de faire découvrir Ant aux développeurs Java qui ne connaissent pas encore cet outil. Afin de dévoiler les principaux apports d'Ant, nous allons comparer la création d'un projet en ligne de commande classique avec la création du même projet avec Ant. ♪

Article lu 19979 fois.

## L'auteur

[Selim KEBIR](https://www.developpez.net/forums/u90040/linkinselim/)[Profil Pro](https://www.developpez.com/user/profil/90040/linkinselim/)[Site personnel](https://skebir.developpez.com/)[Blog](https://blog.developpez.com/linkinselim/)

## L'article

Publié le 1er octobre 2008

[Version PDF](https://skebir.developpez.com/tutoriels/java/ant/ant.pdf)

[ePub](https://skebir.developpez.com/tutoriels/java/ant/ant.epub), [Azw](https://skebir.developpez.com/tutoriels/java/ant/ant.azw) et [Mobi](https://skebir.developpez.com/tutoriels/java/ant/ant.mobi)

## Liens sociaux

[Viadeo](http://www.viadeo.com/shareit/share/?url=https%3A%2F%2Fskebir.developpez.com%2Ftutoriels%2Fjava%2Fant%2F&Title=Prise+en+main+d%27Ant) [Twitter](http://twitter.com/timeline/home?status=https%3A%2F%2Fskebir.developpez.com%2Ftutoriels%2Fjava%2Fant%2F) [Facebook](http://www.facebook.com/sharer.php?u=https%3A%2F%2Fskebir.developpez.com%2Ftutoriels%2Fjava%2Fant%2F) [Share on Google+](https://plus.google.com/share?url=https%3A%2F%2Fskebir.developpez.com%2Ftutoriels%2Fjava%2Fant%2F)

## I. Introduction[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Ant est un projet open source de la fondation Apache écrit en Java qui vise le développement d'un logiciel d'automatisation des opérations répétitives tout au long du cycle de développement logiciel. Il est téléchargeable à l'adresse suivante [http://ant.apache.org](http://ant.apache.org/).

Ant pourrait être comparé au célèbre outil make sous Unix. Il a été développé pour fournir un outil de construction indépendant de toute plate-forme. Ceci est particulièrement utile pour des projets développés sur et pour plusieurs systèmes ou pour migrer des projets d'un système vers un autre. Il est aussi très efficace pour de petits développements.

Ant repose sur un fichier de configuration XML qui décrit les différentes tâches qui devront être exécutées par l'outil. Ant fournit un certain nombre de tâches courantes qui sont codées sous forme de classes Java. Ces tâches sont donc indépendantes du système sur lequel elles seront exécutées. De plus, il est possible d'ajouter ces propres tâches en écrivant de nouvelles classes Java respectant certaines spécifications.

La popularité d'Ant augmente de jour en jour. Sa souplesse, sa puissance et sa simplicité en ont fait l'un des outils les plus populaires dans le monde de Java.

Les environnements de développement intégrés proposent souvent un outil de construction propriétaire qui son généralement moins souple et moins puissant que Ant. Ainsi des plug-ins ont été développés pour la majorité d'entre eux (JBuilder, Forte, Visual Age …) pour leur permettre d'utiliser Ant, devenu un standard de facto.

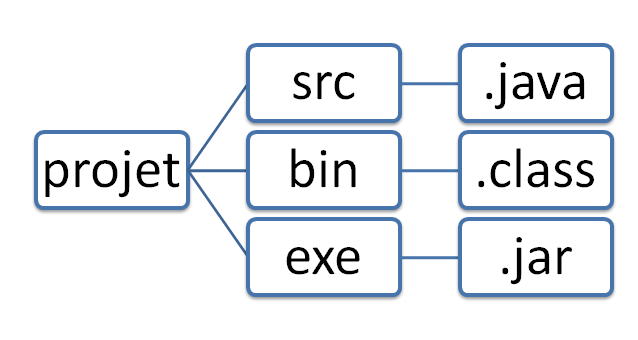
L'IDE Netbeans intègre Ant depuis sa version 4.0, il prévoit un fichier build.xml pour tous ses projets.

## II. Création du projet par ligne de commande[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Dans ce tutoriel, nous voulons séparer les sources des fichiers compilés, donc nos fichiers sources (.java) seront placés dans le répertoire src. Tous les fichiers (.class) générés doivent être placés dans le répertoire bin et enfin le fichier jar de notre projet ira dans le répertoire exe.

Nous avons utilisé la syntaxe de la ligne de commande de Windows. Sous Linux le **md** est remplacé par **mkdir** et l'antislash (\) par un simple slash (/). Les autres commandes devraient toutes marcher sans problème.

On aura donc l'arborescence suivante :

[](https://skebir.developpez.com/tutoriels/java/ant/images/1.PNG)

### II-A. Compilation[▲](https://skebir.developpez.com/tutoriels/java/ant/)

On crée seulement le répertoire src. Voici la commande le permettant :

Sélectionnez

md src

La classe java suivante affiche « Hello world » dans la sortie standard :

HelloWorld.java

Sélectionnez

package org.sdf;

public class HelloWorld {

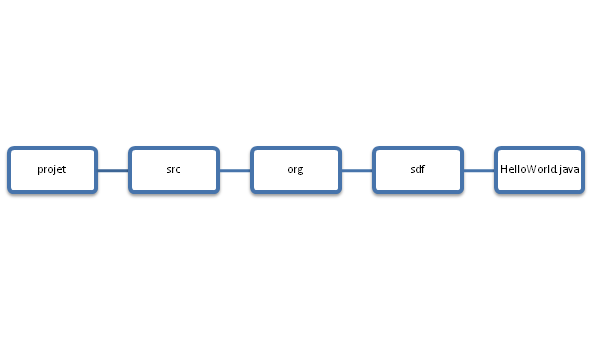
public static void main(String[] args) {

System.out.println("Hello World");

}

}

Vous devez mettre ce code dans un fichier nommé HelloWorld.java. Ce fichier doit se trouver dans le répertoire src/org/sdf. On aura donc ceci :

[](https://skebir.developpez.com/tutoriels/java/ant/images/2.PNG)

Maintenant nous allons essayer de compiler et exécuter notre projet.

Sélectionnez

md bin

javac -sourcepath src -d bin\ src\org\sdf\HelloWorld.java

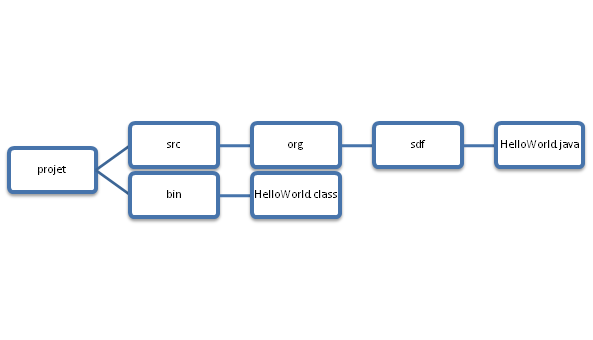
java -cp bin org.sdf.HelloWorld

Qui va résulter en ceci :

Sélectionnez

Hello World.

Après la compilation du fichier source, on aura l'arborescence suivante :

[](https://skebir.developpez.com/tutoriels/java/ant/images/3.PNG)

### II-B. Création du fichier jar exécutable[▲](https://skebir.developpez.com/tutoriels/java/ant/)

La création du fichier jar n'est pas très difficile. Mais le rendre exécutable nécessite plus d'une étape :

1. Création d'un fichier manifest contenant le nom de la class principale ;
2. Création du répertoire exe ;
3. Création du fichier jar.

Tout ceci peut être effectué en ligne de commande comme suit :

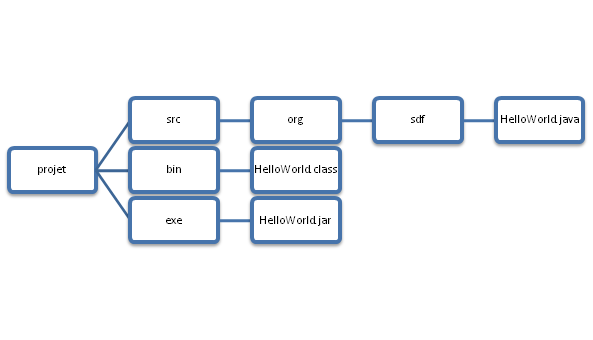
Sélectionnez

echo Main-Class: org.sdf.HelloWorld>MonManifest

md exe

jar cfm exe\HelloWorld.jar MonManifest -C bin .

Après l'exécution de ces trois lignes, on aura l'arborescence de fichiers suivante :

[](https://skebir.developpez.com/tutoriels/java/ant/images/4.PNG)

### II-C. Exécution du projet[▲](https://skebir.developpez.com/tutoriels/java/ant/)

On pourra exécuter notre projet avec la ligne de commande :

Sélectionnez

java -jar exe\HelloWorld.jar

## III. Création du projet avec Ant[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Nous allons maintenant essayer de créer un fichier build.xml Ant. Pour notre projet, ce build.xml contiendra 4 cibles : clean, compile, jar et run.

### III-A. La cible clean[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Cette entrée permet de supprimer tous les fichiers générés lors de la compilation et de la création du fichier. En général, c’est une bonne habitude d'avoir une cible clean.

Sélectionnez

<target name="clean">

<delete dir="bin"/>

<delete dir="exe"/>

</target>

Notre cible clean est très simple, il appellera la tâche delete deux fois, une fois pour supprimer les fichiers class générés, et une autre pour supprimer le fichier jar.

### III-B. La cible compile[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Cette entrée permet de créer un répertoire nommé bin puis de compiler les fichiers sources du répertoire src, et de mettre les fichiers générés dans le répertoire bin.

Sélectionnez

<target name="compile">

<mkdir dir="bin"/>

<javac srcdir="src" destdir="bin"/>

</target>

La création du répertoire bin se fait avec la tâche mkdir. La compilation appelle la tâche javac tout simplement.

### III-C. La cible jar[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Cette entrée permet de créer un répertoire nommé exe et de créer le fichier jar exécutable de notre projet dans ce répertoire.

Sélectionnez

<target name="jar">

<mkdir dir="exe"/>

<jar destfile="exe/HelloWorld.jar" basedir="bin">

<manifest>

<attribute name="Main-Class" value="org.sdf.HelloWorld"/>

</manifest>

</jar>

</target>

La création du répertoire exe se fait avec la tâche mkdir. La création du fichier jar exécutable se fait avec la tâche jar. Le fichier manifest est créé très facilement avec l'élément manifest.

### III-D. Exécution du projet[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Cette entrée permet d'exécuter le fichier jar exécutable généré lors de l'étape précédente.

Sélectionnez

<target name="run">

<java jar="exe/HelloWorld.jar" fork="true"/>

</target>

La tâche java permet d'exécuter le fichier jar. **Fork= « true »** permet d'exécuter le projet dans une nouvelle machine virtuelle Java.

### III-E. Dépendances entre cibles[▲](https://skebir.developpez.com/tutoriels/java/ant/)

L'exécution d'une cible (target) peut dépendre de l'exécution d'autres cibles. Dans notre exemple, la cible jar dépend de la cible compile. Il n'est pas logique de générer un fichier Jar sans que les sources soient compilés.

Ant offre un mécanisme de dépendance qui permet d'ordonner l'exécution des cibles du fichier build.xml. Ce mécanisme est implémenté avec l'attribut depends.

Exemple :

Sélectionnez

<target name="jar" depends="compile">

<mkdir dir="exe"/>

<jar destfile="exe/HelloWorld.jar" basedir="bin">

<manifest>

<attribute name="Main-Class" value="org.sdf.HelloWorld"/>

</manifest>

</jar>

</target>

L'exécution de cette cible requiert l'exécution de la cible **compile**. Ant se chargera de le faire. Si la cible **compile** dépend elle aussi d'une autre cible, alors Ant se chargera d'exécuter les dépendances récursivement.

### III-F. Récapitulatif[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Après avoir écrit les principales cibles de notre projet, nous allons maintenant construire notre fichier build.xml qui sera utilisé par Ant pour exécuter les différentes tâches. Le fichier build.xml doit avoir comme racine un élément appelé **project**. Cet élément peut avoir les attributs suivants :

|  |  |  |
| --- | --- | --- |
| **Attribut** | **Description** | **Obligatoire** |
| name | Le nom du projet | non |
| default | la cible par défaut à utiliser lorsqu'aucune cible n'est indiquée | non |
| basedir | le répertoire de base du projet | non |

Sélectionnez

<project default="run">

<target name="clean">

<delete dir="bin"/>

<delete dir="exe"/>

</target>

<target name="compile" depends="clean">

<mkdir dir="bin"/>

<javac srcdir="src" destdir="bin"/>

</target>

<target name="jar" depends="compile">

<mkdir dir="exe"/>

<jar destfile="exe/HelloWorld.jar" basedir="bin">

<manifest>

<attribute name="Main-Class" value="org.sdf.HelloWorld"/>

</manifest>

</jar>

</target>

<target name="run" depends="jar">

<java jar="exe/HelloWorld.jar" fork="true"/>

</target>

</project>

Maintenant, on peut compiler, archiver et exécuter notre projet avec les commandes :

Sélectionnez

ant clean

ant compile

ant jar

ant run

Et puisque **run** dépend de **jar** qui dépend de **compile** qui dépend de **clean**, On peut faire tout simplement :

Sélectionnez

ant run

Ou mieux encore, puisque la cible par défaut du projet est **run :**

Sélectionnez

ant

### III-G. Amélioration du fichier build.xml[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Maintenant que nous avons notre fichier build.xml, on peut y effectuer quelques optimisations. On remarque qu'on a utilisé plusieurs fois les mêmes noms de répertoires. Un problème surviendrait si l'on voulait changer le répertoire où seront générés les fichiers compilés : il faudrait changer toutes les occurrences de bin dans le fichier.

Pour contourner ce problème, Ant offre le mécanisme de propriétés. Il s'agit de déclarer des propriétés qui ont des valeurs, et de récupérer leur valeur n'importe où dans notre fichier build.xml.

De plus, Ant nous offre la possibilité de garder une trace de l'exécution du fichier build.xml par l'emploi de la tâche **echo**. Ceci peut s'avérer très utile dans certains cas.

build.xml

Sélectionnez

<project default="run">

<property name="src.dir" value="src"/>

<property name="bin.dir" value="bin"/>

<property name="jar.dir" value="exe"/>

<property name="main-class" value="org.sdf.HelloWorld"/>

<target name="clean">

<delete dir="${bin.dir}"/>

<delete dir="${jar.dir}"/>

<echo message="nettoyage termine"/>

</target>

<target name="compile" depends="clean">

<mkdir dir="${bin.dir}"/>

<javac srcdir="${src.dir}" destdir="${bin.dir}"/>

<echo message="compilation terminee"/>

</target>

<target name="jar" depends="compile">

<mkdir dir="${jar.dir}"/>

<jar destfile="${jar.dir}/sdf.jar" basedir="${bin.dir}">

<manifest>

<attribute name="Main-Class" value="${main-class}"/>

</manifest>

</jar>

<echo message="Creation du fichier Jar terminee"/>

</target>

<target name="run" depends="jar">

<java jar="${jar.dir}/sdf.jar" fork="true"/>

</target>

</project>

Maintenant, si on veut changer le répertoire où seront placés les fichiers compilés, il suffira de changer la valeur de la propriété bin.dir.

## IV. Comparaison entre build.xml et la ligne de commande[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Nous avons essayé d'établir un tableau de comparaison entre la compilation, l'archivage et l'exécution en ligne de commande et avec Ant. Ceci est résumé dans le tableau suivant :

|  |  |
| --- | --- |
| **Ligne de commande** | **build.xml** |
| md bin javac -sourcepath src -d bin\ src\org\sdf\HelloWorld.java | <target name=« compile »> <mkdir dir=« bin »/> <javac srcdir=« src » destdir=« bin »/> </target> |
| echo Main-Class: org.sdf.HelloWorld>MonManifest md exe jar cfm exe\HelloWorld.jar MonManifest -C bin . | <target name=« jar »> <mkdir dir=« exe »/> <jar destfile=« exe/HelloWorld.jar » basedir=« bin »> <manifest> <attribute name=« Main-Class » value=« org.sdf.HelloWorld »/> </manifest> </jar> </target> |
| java -jar exe\HelloWorld.jar | <target name=« run »> <java jar=« exe/HelloWorld.jar » fork=« true »/> </target> |

## V. Autres outils[▲](https://skebir.developpez.com/tutoriels/java/ant/)

### V-A. Maven[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Apache Maven est un outil logiciel libre pour la gestion et l'automatisation de production des projets logiciels Java. L'objectif recherché est comparable au système Make sous Unix : produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication.

Il est semblable à l'outil Ant, mais fournit des moyens de configuration plus simples, eux aussi basés sur le format XML. Maven est géré par l'organisation Apache Software Foundation. Précédemment Maven était une branche de l'organisation Jakarta Project.

## VI. Liens utiles[▲](https://skebir.developpez.com/tutoriels/java/ant/)

[Introduction à ANT - partie 1](https://java.developpez.com/articles/ant/partie1/)   
[Introduction à ANT - partie 2](https://java.developpez.com/articles/ant/partie2/)   
[Développons en Java : Ant](http://www.jmdoudoux.fr/java/dej/chap059.htm#chap_59)

## VII. Téléchargement de sources[▲](https://skebir.developpez.com/tutoriels/java/ant/)

[Télécharger](https://skebir.developpez.com/tutoriels/java/ant/fichiers/sources.zip)

## VIII. Conclusion[▲](https://skebir.developpez.com/tutoriels/java/ant/)

En les structurant dans un fichier XML, Ant offre une plus grande lisibilité des opérations qu'on peut exécuter sur un projet. L'intérêt principal est qu'on ne doit plus exécuter les tâches routinières (ennuyantes !!) de compilation, d'archivage et d'exécution à chaque fois qu'on modifie une ligne dans un fichier source du projet : Ant se chargera de faire tout cela pour nous.

## IX. Remerciements[▲](https://skebir.developpez.com/tutoriels/java/ant/)

Un grand merci à [\_Mac\_](https://www.developpez.net/forums/u70411/_mac_/) pour sa relecture attentive. Je remercie aussi [Ricky81](https://www.developpez.net/forums/u24613/ricky81/) pour m'avoir guidé, et [CheryBen](https://www.developpez.net/forums/u62963/cheryben/) pour ses commentaires et critiques très instructifs.

Vous avez aimé ce tutoriel ? Alors partagez-le en cliquant sur les boutons suivants : [Viadeo](http://www.viadeo.com/shareit/share/?url=https%3A%2F%2Fskebir.developpez.com%2Ftutoriels%2Fjava%2Fant%2F&Title=Prise+en+main+d%27Ant) [Twitter](http://twitter.com/timeline/home?status=https%3A%2F%2Fskebir.developpez.com%2Ftutoriels%2Fjava%2Fant%2F) [Facebook](http://www.facebook.com/sharer.php?u=https%3A%2F%2Fskebir.developpez.com%2Ftutoriels%2Fjava%2Fant%2F) [Share on Google+](https://plus.google.com/share?url=https%3A%2F%2Fskebir.developpez.com%2Ftutoriels%2Fjava%2Fant%2F)



[Java 19 : nouvelles fonctionnalités avec exemples, elle apporte des méthodes pour créer des HashMaps préalloués](https://www.developpez.com/voiraussi/clic/i/335380/r/96223/rub/4/)

[Les développeurs de Java SE 6 et 7 évaluent leurs options alors que le support prend fin](https://www.developpez.com/voiraussi/clic/i/335182/r/96139/rub/4/)

[IntelliJ IDEA Conf : l'événement virtuel gratuit pour les développeurs revient pour une nouvelle édition](https://www.developpez.com/voiraussi/clic/i/336303/r/96561/rub/4/)

[JetBrains - Comment travailler avec Amazon Web Services (AWS) dans IntelliJ IDEA, un tutoriel de Dalia Abo Sheasha](https://www.developpez.com/voiraussi/clic/i/333690/r/95527/rub/4/)

[Contacter](https://club.developpez.com/contacts/contribuer) le responsable de la rubrique Java

* [Nous contacter](https://club.developpez.com/contacts/)
* [Participez](https://www.developpez.com/participez/)
* [Hébergement](https://www.developpez.com/hebergement/)
* [Publicité / Advertising](https://www.developpez.com/advertising/)
* [Informations légales](https://www.developpez.com/legal/)
* Partenaire : [Hébergement Web](https://www.planethoster.com/fr/)

© 2000-2022 - www.developpez.com