

## WindowsでAntを使う | 虚ろなるパソコン技術ノート for Windows

ノートブ… 議事録 for iPad

作成日: 2014/10/18 10:49

URL: <http://uturo2.nevernothing.jp/ant/>

---

### WindowsでAntを使う

作成日:2013/03/03 コメント:(0)

カテゴリー: [Java](#), [プログラミング言語](#)

タグ: [Java](#), [Windows7](#)

WindowsにAntをインストールしてつかう。

Ant というのはJavaの開発で使われるビルドツールの一つ。

C言語などの開発でmakeというビルドツールが使われるがそれと似たようなもの。

makeは強力だけど記述が複雑なうえに使われるコマンドがOSのコマンドそのものなので動作がOS依存になってしまい、例えばWindowsとLinuxで共通のmakefileを作るのが難しい。

AntはJavaのコンパイルやファイルのコピー、削除などのコマンドがXMLで記述する独自のものになっており

OS依存性がほとんどない。

OS依存性をなくすかコマンドを新たに覚えなければならないのか

どっちを天秤にかけるかという問題に行き着いてしまうが、

開発をWindowsで行い、本番はLinuxというやり方が大半なのでOS依存がなくなるほうがまあ、助かるといえば助かる。

AntはEclipseの中に含まれているので別に新たに別途インストールする必要もないのだけど

コンソールでしか操作できないLinuxで使うbuildファイルを作成することもあるだろうし、ということでインストールしてみる。

まあ、最近はMavenというツールも使われることがおおいのでどこまで需要があるかなって。

## 1. ダウンロード

Antの開発/維持はThe Apache Software Foundationで行われている。  
もともとはTomcatをビルドするために作られたものらしい。

現在はApache財団のトッププロジェクトになっている。  
ダウンロードサイトは[ここ](#)。

この記事を書いている段階での最新版バージョン1.8.4をダウンロードする。  
Ant自体がJavaで作られているのでWindows版/Linux版などの区別はない。

## 2. インストール

インストールといってもなにも複雑なことはない。  
単にダウンロードした圧縮ファイルを展開してディスク内におきPATHを通すだけ。

ダウンロードしたZIPファイルを展開して中のapache-ant-1.8.4というディレクトリを  
好きな場所、ここでは C:¥ 直下に置き、C:¥Ant とリネームしておく。

環境変数の設定を開き  
ANT\_HOME=C:¥Ant  
を新規作成して  
PATH に  
%ANT\_HOME%¥bin  
を追加する。

コマンドプロンプトを開いて

```
C:\> ant
```

と打ってEnterを押す。

```
Buildfile: build.xml does not exist!  
Build failed
```

と出ればインストール成功。

## 3. ビルドファイルを書いてみる

make では makefile というファイルを作るが  
Antでは build.xml というファイルを作成する。  
もちろん別のファイル名で作って指定することも可能。

なにかプログラムを作ってみる。

C:¥hello¥src¥app というディレクトリを作成してそこにJavaのソースを置く。

```
C:\> cd \  
C:\> mkdir hello  
C:\> cd hello  
C:\> mkdir src  
C:\> cd src  
C:\> mkdir app  
C:\> cd app
```

ここで Hello.java というプログラムファイルを作ってみる。  
内容は以下のもの

Hello.java

```
package app;  
  
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hello");  
    }  
}
```

保存して C:\hello ディレクトリに移動する。

```
C:\> cd \hello
```

ここ C:\hello に build.xml を作成する。

内容は以下のとおり。

build.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>  
<project basedir="." default="build" name="hello">  
    <property environment="env"/>  
    <path id="hello.classpath">  
        <pathelement location="bin"/>  
    </path>  
    <target name="init">  
        <mkdir dir="bin"/>  
    </target>  
    <target name="clean">  
        <delete dir="bin"/>  
    </target>  
    <target depends="init" name="build">  
        <echo message="${ant.project.name}: ${ant.file}"/>  
        <javac debug="true" destdir="bin" includeantruntime="false">  
            <src path="src"/>  
            <classpath refid="hello.classpath"/>  
        </javac>  
    </target>
```

```
</project>
```

みればなんとなくわかるであろうが、

target というのがAntでどんな処理をさせるか指示するもので  
ここでは clean, init, build というのを定義している。

build がデフォルトのtarget になっている。  
そしてbuild を指定したときはその前に必ず init が実行されるようになっている。

```
C:\> ant init
```

とすれば bin というディレクトリが作成され、

```
C:\> ant build
```

もしくは単に

```
C:\> ant
```

とすればinit が実行された後にコンパイルが行われ  
コンパイル結果が bin ディレクトリ内に置かれる。

```
C:\> cd src
```

```
C:\> java app.hello
```

とすればコンパイルされたプログラムが実行される。

```
C:\> ant clean
```

とすれば bin ディレクトリが削除される。

ant コマンドは必ず build.xml が置かれているディレクトリで実行すること。

Antのコマンドをタスクというがファイルのコピーや削除、ディレクトリの作成、  
ファイルの中身を置き換えながらコピー、  
FTPでファイルを転送などいろいろできるタスクが用意されている。

Antのbuild.xmlの書き方の日本語マニュアルは

<http://www.jajakarta.org/ant/>

にある。

残念ながらバージョン1.6で翻訳はとまっているみたい。

でもまあ、大して変わっていないはず。

まあ、あれですね、Eclipseで書いたほうが楽ですよと。

最近ではIvy という必要なjarを自動的にとってくるMavenと同じような機能も開発されているような。

こっちはまだ日本語での情報がすくないのでよくわからない。

以上。

 **2 users**   ツイート 0

