



知识库 .NET技术 Web标准 软件设计 软件工程 数据库 程序人生 求职面试 项目管理 软件测试

新闻 全部 博客 评论 小组 博问



ASP.NET Silverlight Linq JQuery

您的位置: [知识库](#) » 博客文章

## 用Ant构建Java项目

作者: [小言身寸](#) 来源: [BlogJava](#) 发布时间: 2006-08-30 17:36 阅读: 373 次 [原文链接](#) [\[收藏\]](#)

Ant是一个开放源代码的构建工具，专门用于构建Java项目。

Ant入门的最容易的方法就是编写一个构建文件-----build.xml。当Ant运行时，会在当前目录下寻找名为build.xml的文件。

编写Ant构建文件：

第一步：定义项目(**project**)

在项目根目录下创建build.xml文件。第一行指明其为XML格式，然后在第二行定义项目。

```
<?xml version="1.0"?>
```

```
<project name="dms" default="compile" basedir=". ">
```

一个Ant构建文件定义一个项目，使用<project>元素的name属性命名项目，将default属性设为compile，告诉Ant当你在命令行键入ant而无其他参数时，将执行构建过程的compile步骤。我们稍候将定义此步骤；将basedir属性设为"."，告诉Ant，该构建文件内的全部其他路径都是相对于build.xml文件所在路径而言的。

第二步：定义性质(**property**)

下一步，为项目的每一个目录定义性质，以便让Ant了解项目的目录结构。

### Google广告

### 最新新闻

[多图]手机发烧友最爱 谷歌Nexus One手机  
1.8万人签署在线请愿书敦促欧盟保护MySQL  
[视频]谷歌正式发布Nexus One 同时在欧美  
雅虎反驳怀疑论: 1亿美元广告宣传取得巨大成  
赛门铁克产品不能识别2010年 引起混乱  
联通20天卖20万部iPhone  
Google Nexus One将拥有Flash Player

```
<property name="build.dir"      location="build"/>
<property name="build.prod.dir" location="${build.dir}/prod"/>
<property name="build.test.dir" location="${build.dir}/test"/>
<property name="doc.dir"        location="doc"/>
<property name="index.dir"      location="index"/>
<property name="src.dir"        location="src"/>
<property name="test.dir"       location="test"/>
<property name="vendor.lib.dir" location="vendor/lib"/>
```

location属性所指的目录相对于<project>元素定义的basedir属性值（基目录）。

使用Ant性质命名相对目录有两个优点。首先，当Ant运行构建文件时，会定位build.xml文件所在目录的全部相对目录。第二个好处是，性质可以使构建文件易于维护。例如用build.dir性质指代build目录，当以后决定更改build目录为classes时，只要修改build.dir性质的location属性值即可。在构建文件中所有引用build.dir性质的地方，都将会自动指向新的classes目录。

### 第三步：定义路径(path)

下面定义表示项目Java类路径的路径。

```
<path id="project.classpath">
  <pathelement location="${build.prod.dir}" />
  <pathelement location="${build.test.dir}" />
  <fileset dir="${vendor.lib.dir}">
    <include name="*.jar"/>
  </fileset>
</path>
```

<path>元素创建路径project.classpath。我们将在构建过程的编译步骤用到此路径。

首先，用<pathelement>元素向路径加入两个构建目录，以便Java编译器可以找到所有的类文件，然后使用<fileset>元素，加入所有厂家.jar文件至此路径，使Java编译器可以找到我们用到的厂家类。

### 第四步：定义目标(target)

接着我们定义首个构建步骤-----称为“目标”。该目标创建编译步骤将要存放Java类文件的目录。

```
<target name="prepare">
  <mkdir dir="${build.prod.dir}"/>
  <mkdir dir="${build.test.dir}"/>
</target>
```

目标只是定义了一系列任务按照指定的顺序执行的有名字的构建步骤。prepare目标使用<mkdir>任务创

## 热点新闻

[.NET十年回顾：微软目标是否实现？](#)

[2010年1月微软最有价值专家（MVP）当选名超炫Android2.1 谷歌Nexus One界面赏析](#)

["小黑"终结？传联想将推红色ThinkPad](#)

[九城发公开信挖角腾讯用户：玩不爽两亿Q币 熊猫烧香作者出狱赴京 瑞星工程师质疑其水平](#)

[Chrome OS：叫板Windows，超越Linux](#)

## 热门文章

[C#多线程学习\(一\) 多线程的相关概念](#)

[经典推荐--.NET几大热点问题（.NET人员必几个ASP.NET小技巧](#)

[查询表达式\(LINQ\)简介](#)

[LINQ to SQL语句\(1\)之Where](#)

[一步一步学Linq to sql（一）：预备知识](#)

[ASP.NET跨页面传值技巧总结](#)

建构建输出目录，这与在命令行上执行`mkdir`是等效的。但前者可以工作于任何操作系统下，不会因为目录已经存在而失败，而且必要时会创建父目录。

第五步：定义编译步骤

终于我们可以定义编译步骤了。他将编译所有的产品源文件，将结果类文件放到`prepare`目标所生成的目录下。

```
<target name="compile" depends="prepare">
  <javac srcdir="${src.dir}" destdir="${build.prod.dir}">
    <classpath refid="project.classpath" />
  </javac>
</target>
```

通过目标的`depends`属性，可以生成目标间的依存关系，从而指定目标顺序。该属性值是一个目标名间用逗号隔开的清单。

例如，`compile`目标使用`depends`属性来声明与`prepare`目标的依存关系，这意味着要运行`compile`目标必须首先运行`prepare`目标，以在编译源文件前生成要求的目录。与`shell`脚本或批处理文件总是顺序执行命令相比，这是一个很大的优点。

`prepare`目标运行后，`compile`目标运行`<javac>`任务，从而调用Java编译器。编译器处理`src.dir`性质所指目录下的全部Java源文件，在`build.prod.dir`性质所指目录下生成对应的类文件。

`<classpath>`元素告诉编译器使用`project.classpath`代表的值作为类路径。

下面是完整的构建文件：

```
<? xml version="1.0" ?>
<!--
Excerpted from the book, "Pragmatic Project Automation"
ISBN 0-9745140-3-9
Copyright 2004 The Pragmatic Programmers, LLC. All Rights Reserved.
Visit www.PragmaticProgrammer.com
-->

< project name="dms" default="compile" basedir="." >

  < property name="build.dir" location="build" />
  < property name="build.prod.dir" location="${build.dir}/prod" />
  < property name="build.test.dir" location="${build.dir}/test" />
```

```
? < property name ="doc.dir"      location ="doc" />
? < property name ="index.dir"     location ="index" />
? < property name ="src.dir"       location ="src" />
? < property name ="test.dir"      location ="test" />
? < property name ="vendor.lib.dir" location ="vendor/lib" />
?
? < path id ="project.classpath" >
?   < pathelement location ="${build.prod.dir}" />
?   < pathelement location ="${build.test.dir}" />
?   < fileset dir ="${vendor.lib.dir}" >
?     < include name ="*.jar" />
?   </ fileset >
? </ path >
?
? < target name ="prepare" >
?   < mkdir dir ="${build.prod.dir}" />
?   < mkdir dir ="${build.test.dir}" />
? </ target >
?
?
? < target name ="compile" depends ="prepare" >
?   < javac srcdir ="${src.dir}" destdir ="${build.prod.dir}" >
?     < classpath refid ="project.classpath" />
?   </ javac >
? </ target >
? </ project >
```

运行构建:

在命令行环境中, 切换至build.xml所在的目录, 运行Ant:

D:\MyProject> ant

运行Ant前, 请确认ANT\_HOME环境变量指向你的Ant安装目录, 且PATH环境变量中加入了%ANT\_HOME%\bin。

Ant读取build.xml, 以预先定义的顺序执行构建步骤(目标)。

