Ant本意是Another Neat Tool,也就是另一种整洁的工具，取首字符就是Ant。

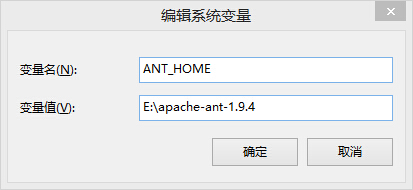
Apache Ant,是一个将[软件](http://baike.baidu.com/view/37.htm" \t "_blank)编译、测试、部署等步骤联系在一起加以自动化的一个工具，大多用于Java环境中的软件开发

# 下载并配置Ant包

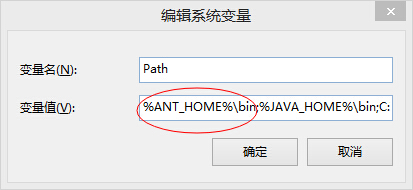
地址：<http://ant.apache.org/bindownload.cgi>

配置Ant环境变量：

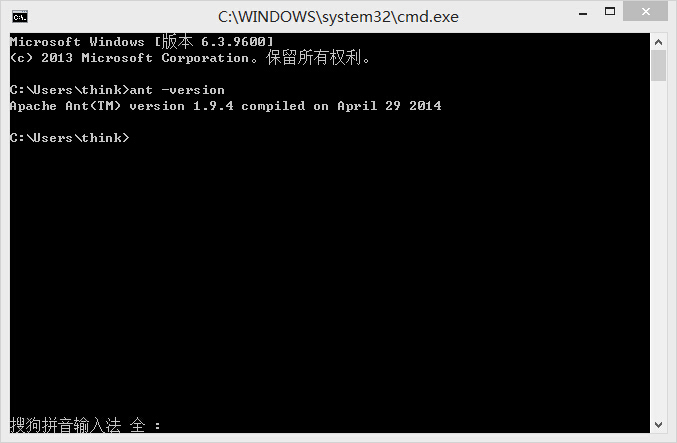
Ant 环境配置：  
Ant解压目录：E:\ apache-ant-1.9.4  
在“系统”－》“高级”－》“环境变量”－》“系统参数”中  
1、新增ANT\_HOME：  
变量名：ANT\_HOME  
变量值：E:\apache-ant-1.9.4



2、在Path变量中增加Ant的bin目录  
已存在的变量名：Path  
在最前面新增的值：%ANT\_HOME%\bin;



3、一路“确定”，关闭所有窗口，即配置完成。打开一个命令行窗口（在此之前已打开的命令行窗口不能生效），运行如下命令：ant -version。即可打印出ant的版本信息，如果出现下面的画面，表明Ant环境已经配置好了。



# 例1 命令行运行ant

1. 在D:\admin下新建 build.xml

备注：没有指定任何参数时，Ant会在当前目录下查询build.xml文件，如果其他文件名可用ant –buildfile a.xml

<?xml version="1.0" encoding="UTF-8"?>

<project name ="antPro" default ="getBaseDir" basedir ="D:/admin">

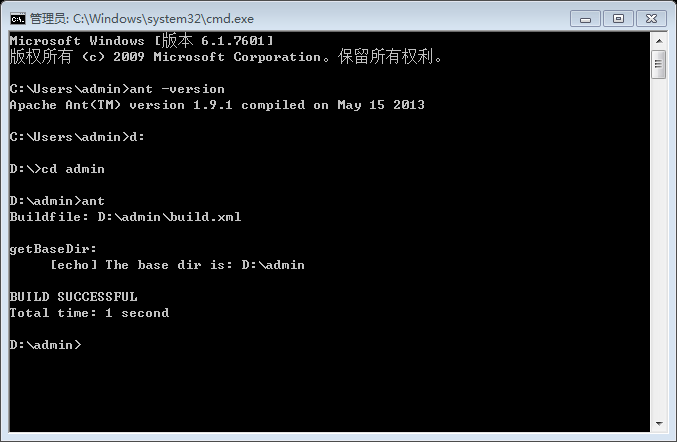
<target name="getBaseDir">

<echo message="The base dir is: ${basedir}"/>

</target>

</project>

2、命令行进入 D:\admin ，输入ant 出现如下结果



# 例2 构建文件的名字不是build.xml

则Ant运行的时候就可以使用–buildfile file ,这里file 指定了要使用的构建文件的名称 ，使用属性信息

ant –buildfile build1.xml 或者 ant –file build1.xml

build1.xml

<?xml version="1.0" encoding="UTF-8"?>

<project name="antProperty" default="printPro" basedir=".">

<property name="name" value="Kevin"/>

<property name="age" value="25"/>

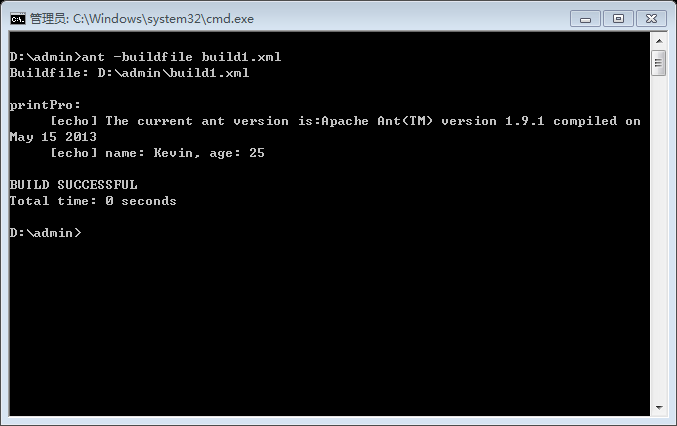
<target name="printPro">

<echo message="The current ant version is:${ant.version}"/>

<echo message="name: ${name}, age: ${age}"/>

</target>

</project>



# 例3 执行指定的target

build2.xml

<?xml version="1.0" ?>

<project name ="antPro" default ="A" >

<target name="A" depends="B">

<echo message="AAA"/>

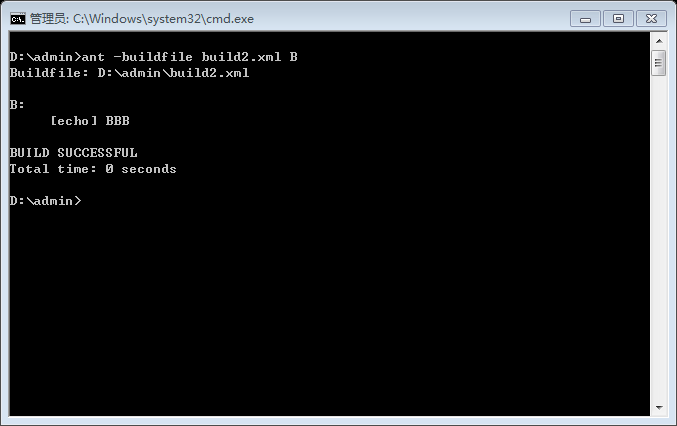
</target>

<target name="B">

<echo message="BBB"/>

</target>

</project>



# 5、ant关键元素

## project元素

    project 元素是 Ant 构件文件的根元素， Ant 构件文件至少应该包含一个 project 元素，否则会发生错误。在每个 project 元素下，可包含多个 target 元素。接下来向读者展示一下 project 元素的各属性。

**1) name 属性**

    用于指定 project 元素的名称。

**2) default 属性**

    用于指定 project 默认执行时所执行的 target 的名称。

**3) basedir 属性**

 用于指定基路径的位置。该属性没有指定时，使用 Ant 的构建文件的父目录作为基准目录。

basedir="." 指的是当前路径

## ****target 元素****

   target为Ant的基本执行单元，它可以包含一个或多个具体的任务。多个target 可以存在相互依赖关系。它有如下属性：

**1) name 属性**

    指定 target 元素的名称，这个属性在一个 project 元素中是唯一的。我们可以通过指定 target 元素的名称来指定某个 target 。

**2) depends 属性**

用于描述 target 之间的依赖关系，若与多个 target 存在依赖关系时，需要以“,”间隔。 Ant 会依照 depends 属性中 target 出现的顺序依次执行每个 target 。被依赖的 target 会先执行。

一个target只能被执行一次，即时有多个target依赖于它

<target name="A"/>  
<target name="B" depends="A"/>  
<target name="C" depends="B"/>  
<target name="D" depends="C,B,A"/

**3) if 属性**

    用于验证指定的属性是否存在，若不存在，所在 target 将不会被执行。

**4) unless 属性**

    该属性的功能与 if 属性的功能正好相反，它也用于验证指定的属性是否存在，若不存在，所在 target 将会被执行。

**5) description 属性**

该属性是关于 target 功能的简短描述和说明

<?xml version="1.0" ?>

<project name ="targetPro" default="targetB">

<target name="targetA" if ="ant.java.version">

<echo message ="Java Version: ${ant.java.version}"/>

</target>

<target name="targetB" depends ="targetA" unless ="philander">

<description>

a depend example!

</description>

<echo message ="The base dir is: ${basedir}"/>

</target>

</project>

## ****property 元素****

property元素可看作参量或者参数的定义

<property name="name" value="kevin"/>

<echo message="name:${name}"/>

# 6、Ant常用命令

## ****1. copy 命令****

copy主要用来对文件和目录的复制功能。举例如下：

1.拷贝单个的文件：

<copy file="myfile.txt" tofile="mycopy.txt"/>

2.拷贝单个的文件到指定目录下

<copy file="myfile.txt" todir="../some/other/dir"/>

3.拷贝一个目录下的文件到另外一个目录下

<copy todir="../mycode1/testngPara">

<fileset dir="../admin"/>

</copy>

4.拷贝一批指定的文件到指定目录下

<copy todir="../dest/dir">

<fileset dir="src\_dir">

<include name="\*\*/\*.java"/>

<exclude name="\*\*/Test.java"/>

</fileset>

</copy>

或者

<copy todir="../dest/dir">

<fileset dir="src\_dir" excludes="\*\*/\*.java"/>

</copy>

## ****2. delete**** 命令

对文件或目录进行删除，举例如下：

eg1. 删除某个文件：

<delete file="/home/photos/philander.jpg"/>

eg2. 删除指定目录及其子目录

<delete dir="../home/photos"/>

eg3.删除指定的一组文件

<delete>

<fileset dir="." includes="\*\*/\*.txt"/>

</delete>

## ****3. mkdir**** 命令

创建目录 eg ： <mkdir dir="./tom/tom123" />

## ****4. move**** 命令

移动文件或目录，举例如下：

eg1. 移动单个文件：

<move file="a.txt" tofile="aaa.txt"/>

eg2. 移动单个文件到另一个目录：

<move file="b.txt" todir="./jerry" />

eg3. 移动某个目录到另一个目录：（即重命名文件名）

<move todir="kevin1"> <fileset dir="tom"/> </move>

## 5****. echo**** 命令

 该任务的作用是根据日志或监控器的级别输出信息。

<echo message="Hello" file="./ant.log" append="true" />

# 7、利用ant构建和部署java工程

1）建立名为 JavaTestPro的Java项目，包 com.edu.anttest建立HelloWorld.java这个类文件。该类文件的内容如下：

public class HelloWorld {

public static void main(String[] args) {

System.out.println("hello world!");

}

}

2) 在JavaTestPro项目的根目录下建立build.xml 文件

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

<project name =*"javacTest"* default=*"run"* basedir=*"."*>

<target name=*"clean"*>

<delete dir=*"${basedir}/build"*/>

</target>

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

<mkdir dir =*"${basedir}/build/classes"*/>

<javac srcdir =*"${basedir}/src"* destdir =*"${basedir}/build/classes"* includeantruntime=*"on"* encoding="utf-8"/>

</target>

<target name=*"run"* depends =*"compile"*>

<java classname =*"com.ant.test.HelloWorld"*>

<classpath>

<pathelement path=*"${basedir}/build/classes"*/>

</classpath>

</java>

</target>

</project>

或者采用全局变量

<?xml version="1.0" ?>

<project name ="javacTest" default="run" basedir=".">

<property name="dest" value="build/classes" />

<target name="clean">

<delete dir="${dest}"/>

</target>

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

<mkdir dir ="${dest}"/>

<javac srcdir ="src" destdir ="${dest}" includeantruntime="on"/>

</target>

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

<java classname ="com.ant.test.HelloWorld">

<classpath>

<pathelement path="${dest}"/>

</classpath>

</java>

</target>

</project>

3) 选中项目 右键 run as >ant build

编译后生成的HelloWorld.class文件就在该目录下

# 8、ant+testng

{classpath}的路径，跟着lib目录下的所有jar文件，接着是classes目录

<?xml version="1.0" ?>

<project name ="javacTest" default="run" basedir=".">

<!-- 定义全局变量 -->

<property name="dest" value="./build/classes" />

<property name="testng.output.dir" value="./test-output"/>

<!-- 定义类路径 -->

<path id="classes">

<fileset dir="lib" includes="\*.jar"/>

<pathelement location="${dest}"/>

</path>

<taskdef resource="testngtasks" classpath="lib/testng-6.8.jar"/>

<target name="clean">

<delete dir="${dest}"/>

</target>

<!-- 编译 src目录 -->

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

<mkdir dir ="${dest}"/>

<javac srcdir="src" destdir="${dest}" classpathref="classes" includeantruntime="on" />

</target>

<!-- 运行 -->

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

<testng outputdir="${testng.report.dir}" classpathref="classes" >

<xmlfileset dir="." includes="testng.xml"/>

</testng>

</target>

</project>

# 9、定时任务设置

1. 在d:/admin 新建test.bat，输入

d:

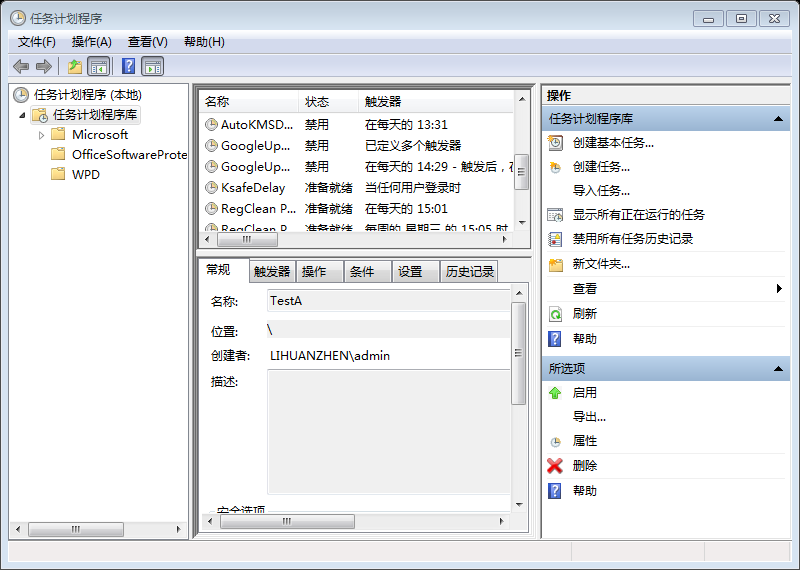
cd admin

echo %date% %time% hello world >> a.txt

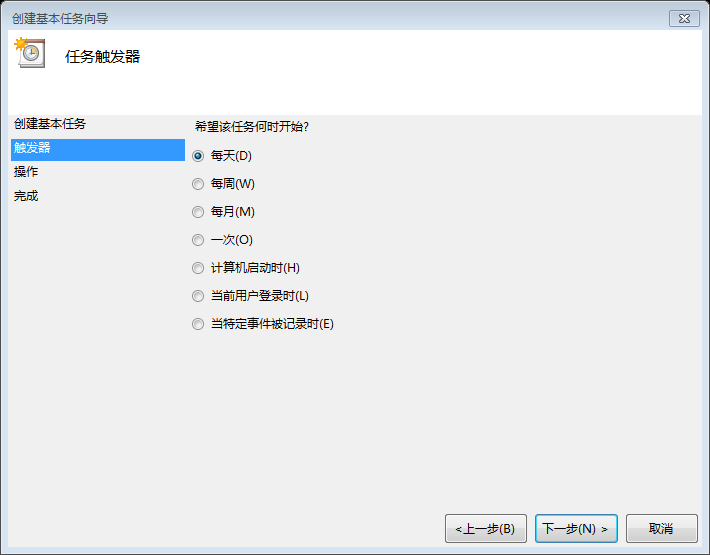
pause on

1）控制面板>管理工具>任务计划程序

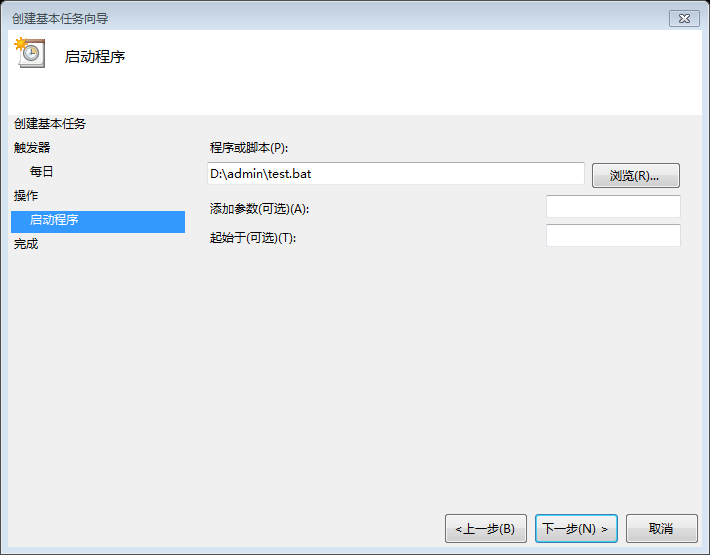
创建基本任务”testB”



2）触发器 选择”每天”

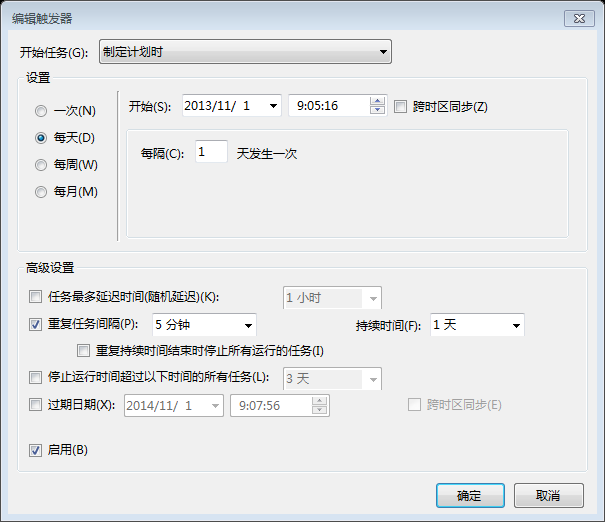


3）启动程序 选择 D:\admin\test.bat



4）修改 testB的属性信息

每隔五分钟运行一次



# 10、定时任务与ANT

Test1.bat内容如下

call ant

pause

接下来 请为 Test1.bat 制作为定时任务

<?xml version="1.0" ?>

<project name ="javacTest" default="run" basedir=".">

<!-- 定义全局变量 -->

<property name="dest" value="./build/classes" />

<property name="testng.output.dir" value="./test-output"/>

<!-- 定义类路径 -->

<path id="classes">

<fileset dir="lib" includes="\*.jar"/>

<pathelement location="${dest}"/>

</path>

<taskdef resource="testngtasks" classpath="lib/testng-6.8.jar"/>

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

<testng outputdir="${testng.report.dir}" classpathref="classes" >

<xmlfileset dir="." includes="testng.xml"/>

</testng>

</target>

<target name="clean">

<delete dir="${basedir}/build"/>

</target>

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

<mkdir dir ="${dest}"/>

<javac srcdir ="${basedir}/src" destdir ="${dest}" includeantruntime="on"/>

</target>

</project>

<javac srcdir="src" destdir="build/classes">

<compilerarg line="-encoding UTF-8"/> </javac>