

# Agile Java

阿左<sup>1</sup>      Nobody<sup>2</sup>

October 8, 2012

<sup>1</sup>感谢读者  
<sup>2</sup>感谢国家

# Contents

<b>I 基础入门</b>	<b>4</b>
<b>1 开发环境</b>	<b>5</b>
1.1 开发环境配置 . . . . .	5
1.2 编写第一个 ant 程序 . . . . .	5
<b>2 起步</b>	<b>7</b>
2.1 project 元素 . . . . .	7
2.2 target 元素 . . . . .	8
2.2.1 depends 属性 . . . . .	8
2.2.2 if 属性 . . . . .	8
2.2.3 unless 属性 . . . . .	9
2.3 property 元素 . . . . .	10
2.4 task 元素 . . . . .	11
2.5 命令行使用 ant . . . . .	11
2.5.1 指定文件 . . . . .	11
2.5.2 显示任务描述 . . . . .	11
2.5.3 指定变量的值 . . . . .	11

<b>3 核心类型</b>	<b>12</b>
3.1 断言类型: Assertions Type . . . . .	12
3.2 匹配模式: PatternSet . . . . .	13
3.2.1 基本属性 . . . . .	13
3.2.2 例子 . . . . .	14
 <b>II 工具整合</b>	 <b>15</b>
 <b>III 实践</b>	 <b>16</b>

# **List of Figures**

# **List of Tables**

# Part I

## 基础入门

# Chapter 1

## 开发环境

### 1.1 开发环境配置

基本环境变量的配置：

JAVA\_HOME、ANT\_HOME、PATH 这三个环境变量要配置正确。

### 1.2 编写第一个 ant 程序

```
1 <?xml version="1.0"?>
2
3 <project name="myproject" default="firstExample">
4
5   <target name="firstExample">
6     <echo message="This is my first targe" />
7     <echo message="          os: ${os.name}" />
8     <echo message="          baseidr: ${basedir}" />
9     <echo message=" ant fileis: ${ant.file}" />
```

```
10     <echo message=" ant version: ${ant.version}" />
11     <echo message="project name: ${ant.project.name}" />
12     <echo message="java version: ${ant.java.version}" />
13 </target>
14
15 </project>
```



# Chapter 2

## 起步

### 2.1 project 元素

一个 ant 文件中一定要有一个 project 元素。name 属性定义工作名字；default 属性代表默认启动 target；description 属性定义说明文本。

```
1 <?xml version="1.0"?>
2
3 <project name="myproject" default="firstExample">
4
5   <description>
6     Show this description with command:
7     ant -projecthelp
8   </description>
9
10 </project>
```

## 2.2 target 元素

target 元素一定要有 name 属性。

### 2.2.1 depends 属性

表示当前 target 执行依赖于其他的 target 成功执行。可以有多个 depends 之间用“,”分隔。

```
1 <?xml version="1.0"?>
2
3 <project name="myproject" default="firstExample">
4
5   <description>
6     Show this description with command:
7     ant -projecthelp
8   </description>
9
10 </project>
```

### 2.2.2 if 属性

验证执行前必须设定的属性。例如设定 JDK:

```
1 <?xml version="1.0"?>
2
3 <!--
4   ant -f build.03.xml
5   ant -f build.03.xml checkFinished
6 -->
7 <project name="myproject" default="firstExample">
8
```

```
9 <target name="showJava" if="ant.java.version">
10   <echo message="java version: ${ant.java.version}" />
11 </target>
12
13 <!--
14 <property name="isFinished" value="finished" />
15 -->
16
17 <target name="checkFinished" if="isFinished">
18   <echo message="is finished: ${isFinished}" />
19 </target>
20
21 <target name="firstExample" depends="showJava, checkFinished">
22   <echo message="always show this" />
23 </target>
24 </project>
```

### 2.2.3 unless 属性

没有设置才执行

```
1 <?xml version="1.0"?>
2
3 <!--
4   ant -f build.04.xml checkUnFinished
5 -->
6 <project name="myproject" default="firstExample">
7   <!--
8   <property name="isFinished" value="finished" />
9   -->
10
11   <target name="checkUnFinished" unless="isFinished">
12     <echo message="show this if not set ifFinished" />
```

```
13 </target>
14
15 <target name="firstExample" depends="checkUnFinished">
16     <echo message="always show this" />
17 </target>
18
19 </project>
```

## 2.3 property 元素

property 可以看作参数或是变量的定义。可以在构建文件中通过 property 元素建立，也可以在构建文件外建立一个 build.properties 文件来存放。

```
1 <?xml version="1.0"?>
2
3 <project name="myproject" default="firstExample">
4
5     <!-- reference properties file -->
6     <property file="build.properties" />
7
8     <property name="src" value="src" />
9
10    <target name="init">
11        <!-- get current time -->
12        <tstamp/>
13    </target>
14
15    <target name="firstExample" depends="init">
16        <echo message="author name:${author.name}" />
17        <echo message="author nick:${author.nickname}" />
18        <echo message="currentTime:${DSTAMP}" />
19    </target>
```

```
20  
21 </project>
```

## 2.4 task 元素

task 元素是一系列完成指定功能的脚本和程序。比如自带的编译用的 `javac`、打包用的 `jar`、建立目录的 `mkdir` 等。用户也可以编写自己的 task。

## 2.5 命令行使用 ant

### 2.5.1 指定文件

`-buildfile` 或 `-file` 或 `-f` 都可以。

### 2.5.2 显示任务描述

`-projecthelp`。

### 2.5.3 指定变量的值

`-D<property>=<value>`

# Chapter 3

## 核心类型

### 3.1 断言类型: Assertions Type

断言类型指定哪些类需要让 ant 工具执行 java 断言。enableSystemAssertions 属性设定是否允许系统断言，默认为 unspecified。

断言类型内还可以定义 enable 类型与 disable 类型，通过这两个类型的 class 属性与 package 属性来设置指定的类或是包是否可以执行断言。

允许所有的用户代码执行断言：

```
1 <assertions>
2   <enable />
3 </assertions>
```

只允许 Test 这个类执行断言：

```
1 <assertions>
2   <enable class="Test"/>
3 </assertions>
```

允许一个包下所有的类都执行断言：

```
1 <assertions>
2   <enable package="org.apache"/>
3 </assertions>
```

更复杂的定义：

```
1 <assertions>
2   <enable package="org.apache"/>
3   <disable package="org.apache.tools.ant"/>
4   <enable class="org.apache.tools.ant.Main"/>
5 </assertions>
```

引用一个已经存在的定义：

```
1 <assertions id="project.assertions">
2   <disable package="org.apache.test"/>
3 </assertions>
4
5 <assertions refid="project.assertions" />
```

## 3.2 匹配模式：PatternSet

### 3.2.1 基本属性

匹配格式有以下四个属性：

includes 属性：可以指定多个模式用逗号或空格分隔，相当于一个列表。

includesfile 属性：指定具体包含的文件。可以引用在 properties 文件中指定的内容。

excludes 属性：和上面的相反。

excludesfile 属性：和上面的相反。

以上四个属性都有三个属性来匹配指定的内容：

name 属性：文件名、路径或格式。

if 属性：指定的属性有定义则生效。

unless 属性：指定的属性没有定义则生效。

### 3.2.2 例子

类名中不包含 Test 的类：

```
1 <patternset id="non.test.sources">
2   <include name="**/*.java" />
3   <exclude name="**/*Test.java" />
4 </patternset>
```



## Part II

### 工具整合

## Part III

### 实践