

|  |
| --- |
|  |

**Apache Ant Guidelines and Best Practices**

**Document History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version No. | Authored/Modified by | Modification Date | Reviewed by, Date | Remark / Change history |
| 1.0 | Abhijeet Kamble | 15th July 2015 |  | Initial doc |
| 1.1 | Aparna Nigdikar | 21st July 2015 |  | Added document history section |

# Table of Contents

[Table of Contents 3](#_Toc425961535)

[1. Introduction 4](#_Toc425961536)

[2. COnfiguration Guidelines FOr ANT 4](#_Toc425961537)

[3. USer Guidelines FOr ANT 5](#_Toc425961538)

[4. ANT guidelines 6](#_Toc425961539)

# 

# Introduction

**Apache Ant** byApache Software Foundation

Build tools are used for Build Automation.

**Build automation** is the act of scripting or automating a wide variety of tasks including:

1. compiling source code into binary code

2. packaging binary code.

3. running tests

4. deployment to production systems

5. creating documentation and/or release note.

According to Ant's original author, James Duncan Davidson. The name is an acronym for "Another Neat Tool".

Apache Ant is a Java library and command-line tool build tool. It drives processes described in build files as targets and extension points dependent upon each other. Ant supplies a number of built-in tasks allowing to compile, assemble, test and run Java and other applications.  
   
Ant is Open Source and implemented /written in Java.   
  
Ant can also be used effectively to build non Java applications, for instance PHP, C or C++ applications. More generally, Ant can be used to pilot any type of process which can be described in terms of targets and tasks.  
Ant can easily integrate with continuous integration (CI) tools (Jenkins, Bamboo).  
  
Users of Ant can develop their own "antlibs" containing Ant tasks and types, and are offered a large number of ready-made commercial or open-source "antlibs".

# COnfiguration Guidelines FOr ANT

Following are the guidelines for developers:

1. Builds run on both Windows and UNIX/Linux systems.
2. Java must installed on system, version 1.5 or later required 1.6 or later strongly recommended.
3. If only the JRE is present but not a full JDK then many tasks will not work.
4. Ensure that the JAVA\_HOME environment variable is set to the folder where your JDK is installed.
5. Create a new environment variable called **ANT\_HOME** that points to the Ant installation folder.
6. To verify the successful installation of Apache Ant type ant on your command prompt.

C:\>ant -version

# USer Guidelines FOr ANT

Following are the user guidelines for Ant:

1. Ant is extremely flexible and does not impose coding conventions or directory layouts.
2. Build targets can still do OS-specific tasks.
3. Works with anything that can be done from the command line.

4. Ant can get source code from version control:-

CVS, Subversion, Synergy, Perforce, Clear Case and many more.

1. Ant can compile source code.
2. The XML element **project** has three attributes (build.xml file) :

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| name | The Name of the project. (Optional) |
| default | The default target for the build script. A project may contain any number of targets. This attribute specifies which target should be considered as the default. (Mandatory) |
| basedir | The base directory (or) the root folder for the project. (Optional) |

1. Ant can run unit tests:-

JUnit3, JUnit4, TestNG, or any arbitrary test application.

**Example:**

<target name="unittest">

<junit haltonfailure="true" printsummary="true">

<test name="com. UtilsTest"/>

</junit>

</target>

1. Ant can package compiled code and resources:-

jars, wars, ears, tars, zips, whatever.

**Example:**

<target name="build-jar">

<jar destfile="${web.dir}/lib/util.jar"

basedir="${build.dir}/classes"

includes="faxapp/util/\*\*"

excludes="\*\*/Test.class">

<manifest>

<attribute name="Main-Class" value="com.util.FaxUtil"/>

</manifest>

</jar>

</target>

# ANT guidelines

Following are the guidelines for Ant User:

1. The Begin and End tags for project (<project> and </project>) MUST start and end the file.
2. The Begin <project> MUST have an attribute called default which is the name of one of the targets.
3. Each build file must have at least one target.
4. The Begin and End tags for <target> and </target> must also match EXACTLY.
5. Each target MUST have a name.
6. Target depends are optional.

The depends attribute can be included in the target tag to specify that this target requires another target to be executed prior to being executed itself. Multiple targets can be specified and separated with commas.

<target name="one" depends="two, three">

Here, target "one" will not be executed until the targets named "two" and "three" are, first.

**Example:**

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

<target name="one">

<echo>Running One</echo>

</target>

<target name="two" depends="one"> <echo>Running Two</echo>

</target>

<target name="three" depends="two"> <echo>Running Three</echo>

</target>

</project>

1. Ant uses the **property** element which allows to specify properties.   
   This allows the properties to be changed from one build to another or from one environment to another.  
   Ant allowed declaring variables such as project name, project source directory, etc.

Properties do not have to be used only inside a target. They can be set anywhere in a build file (or an external property file) and referenced anywhere in a build file after they are set.

1. Wildcards are used by ant to specify groups of files that have a pattern to their names.

-**?** is used to match any character.

-**\***  is used to match zero or more characters.

-**\*\*** is used to match zero or more directories.

**Example:**

<filesetdir="${server.src}" casesensitive="yes"> <include name="\*\*/\*.java"/>

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

</fileset>

1. Anything between <echo> and </echo> tags is outputted to the console if the surrounding target is called.
2. You can execute this from a DOS or UNIX command prompt by creating a file called build.xml and typing: -Ant
3. Ant will search for the build file in the current directory and run the build.xml file.

**Example:**

<? xml version="1.0"?>

<project name="MyFirstAntProject” default="MyTarget">

<target name="init">

<echo>Running target init</echo>

</target>

<target name="MyTarget" depends="init">

<echo>Running target MyTarget</echo> </target>

</project>