



**<APACHE ANT>**

# **LEARN APACHE ANT**

project build management

**tutorialspoint**  
SIMPLY EASY LEARNING

[www.tutorialspoint.com](http://www.tutorialspoint.com)



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

## About the Tutorial

---

Apache ANT is a Java based build tool from Apache Software Foundation. Apache ANT's build files are written in XML and they take advantage of being open standard, portable and easy to understand.

This tutorial will teach you how to use Apache ANT to automate the build and deployment process in simple and easy steps. After completing this tutorial, you will find yourself at a moderate level of expertise in using Apache ANT from where you can take yourself to next levels.

## Audience

---

This tutorial is prepared for the beginners to help them understand basic functionality of Apache ANT tool to automate the build and deployment process.

## Prerequisites

---

We assume you have knowledge of software development using any programming language, especially Java, and the software build and deployment process.

## Disclaimer & Copyright

---

© Copyright 2014 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at [contact@tutorialspoint.com](mailto:contact@tutorialspoint.com).

# Table of Contents

---

About the Tutorial.....	i
Audience .....	i
Prerequisites .....	i
Disclaimer & Copyright.....	i
Table of Contents .....	ii
<b>1. ANT INTRODUCTION.....</b>	<b>1</b>
<b>Need for a Build Tool.....</b>	<b>1</b>
<b>History of Apache ANT .....</b>	<b>1</b>
<b>Features of Apache ANT .....</b>	<b>1</b>
<b>2. ANT ENVIRONMENT SETUP .....</b>	<b>3</b>
<b>Installing Apache ANT .....</b>	<b>3</b>
<b>Verifying Apache ANT Installation .....</b>	<b>3</b>
<b>Installing Eclipse.....</b>	<b>4</b>
<b>3. ANT BUILD FILES.....</b>	<b>5</b>
<b>4. ANT PROPERTY TASK .....</b>	<b>8</b>
<b>5. ANT PROPERTY FILES .....</b>	<b>10</b>
<b>build.xml .....</b>	<b>10</b>
<b>build.properties .....</b>	<b>11</b>
<b>6. ANT SERVICES.....</b>	<b>13</b>
<b>Fileset .....</b>	<b>13</b>
<b>Patternset .....</b>	<b>13</b>
<b>Filelist.....</b>	<b>14</b>
<b>Filterset.....</b>	<b>14</b>
<b>Path .....</b>	<b>15</b>

7. ANT BUILDING PROJECTS.....	16
8. ANT BUILD DOCUMENTATION.....	20
<b>Attributes</b> .....	<b>20</b>
<b>Putting it All Together</b> .....	<b>20</b>
9. ANT CREATING JAR FILES.....	23
10. ANT PACKAGING APPLICATIONS.....	29
<b>build.properties</b> .....	<b>29</b>
<b>build.xml</b> .....	<b>29</b>
11. ANT FOR DEPLOYING APPLICATIONS.....	33
<b>build.properties</b> .....	<b>33</b>
<b>build.xml</b> .....	<b>33</b>
12. EXECUTING JAVA CODE .....	40
13. ECLIPSE INTEGRATION.....	42
14. ANT JUNIT INTEGRATION.....	44
15. EXTENDING ANT.....	46

# 1. ANT INTRODUCTION

ANT stands for Another Neat Tool. Before going into details of Apache ANT, you must understand the need for a build tool.

## Need for a Build Tool

---

Apart from coding and testing, the JAVA developers are engaged in the following tasks:

- Compiling the code
- Packaging the binaries
- Deploying the binaries to the test server
- Testing the changes
- Copying the code from one location to another

To automate and simplify the above tasks, Apache ANT is useful. It is an operating system build and deployment tool that can be executed from command line.

## History of Apache ANT

---

- ANT was created by James Duncan Davidson, the original author of Tomcat.
- It was originally used to build Tomcat, and was bundled as part of Tomcat distribution.
- ANT was born out of the problems and complexities associated with Apache Make tool.
- ANT was promoted as an independent project in Apache in the year 2000.
- The latest version of Apache ANT as on May 2014 is 1.9.4.
- NAnt is a .NET build tool that is similar to ANT, except that is used to build .NET apps.

## Features of Apache ANT

---

- ANT is the most complete Java build and deployment tool available.



- ANT is platform neutral and can handle platform-specific properties such as file separators.
- ANT can be used to perform platform-specific tasks such as modifying the modified time of a file using *'touch'* command.
- ANT scripts are written using plain XML. If you are already familiar with XML, you can learn Ant pretty quickly.
- ANT is good at automating complicated repetitive tasks.
- ANT comes with a big list of predefined tasks.
- ANT provides an interface to develop custom tasks.
- ANT can be easily invoked from the command line and it can integrate with free and commercial IDEs.

## 2. ANT ENVIRONMENT SETUP

Apache ANT is distributed under the Apache Software License, a full-fledged open source license certified by the open source initiative.

The latest Apache Ant version, including full-source code, class files and documentation can be found at <http://ant.apache.org>.

### Installing Apache ANT

---

It is assumed that you have already downloaded and installed Java Development Kit (JDK) on your computer. If not, please follow the instructions [here](#).

- Ensure that the JAVA\_HOME environment variable is set to the folder where your JDK is installed.
- Download the binaries from <http://ant.apache.org>
- Unzip the .zip file to a convenient location using Winzip, winRAR, 7-zip or similar tools, on say c:\folder.
- Create a new environment variable called **ANT\_HOME** that points to the ANT installation folder, in this case, **c:\apache-ant-1.9.4-bin** folder.
- Append the path to the Apache ANT batch file to the PATH environment variable. In our case, this would be the **c:\apache-ant-1.9.4-bin\bin** folder.

### Verifying Apache ANT Installation

---

To verify the successful installation of Apache ANT on your computer, start command prompt and type ant. You should see an output similar to:

```
C:\>ant -version  
Apache Ant(TM) version 1.9.4 compiled on December 20 2014
```

If you do not see the above reply, then please confirm if you followed the installation steps properly.

## Installing Eclipse

---

This tutorial also covers integration of ANT with Eclipse IDE. Hence, if you have not installed Eclipse already, please download and install Eclipse as given below:

- Download the latest Eclipse binaries from [www.eclipse.org](http://www.eclipse.org).
- Unzip the Eclipse binaries to a convenient location, say c:\.
- Run Eclipse from c:\eclipse\eclipse.exe.



# 3. ANT BUILD FILES

Typically, ANT's build file called **build.xml** should reside in the base directory of the project. Although, you are free to use other file name or place for the build file.

For this exercise, create a file called build.xml anywhere in your computer with the following contents in it:

```
<?xml version="1.0"?>
  <project name="Hello World Project" default="info">
    <target name="info">
      <echo>Hello World - Welcome to Apache Ant!</echo>
    </target>
  </project>
```

Note that there must be no blank line(s) or whitespace(s) before the xml declaration. If you allow them, the following error message occurs while executing the ant build

-

*The processing instruction target matching "[xX][mM][lL]" is not allowed.*

All build files require the **project** element and at least one **target** element.

The XML element **project** has three attributes:

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)

A target is a collection of tasks that you want to run as one unit. In our example, we have a simple target to provide an informational message to the user.

Targets can have dependencies on other targets. For example, a **deploy** target may have a dependency on the **package** target, the **package** target may have a dependency on the **compile** target, and so forth. Dependencies are denoted using **depends** attribute. For example:

```
<target name="deploy" depends="package">
    ....
</target>
<target name="package" depends="clean,compile">
    ....
</target>
<target name="clean" >
    ....
</target>
<target name="compile" >
    ....
</target>
```

The target element has the following attributes:

Attributes	Description
name	The name of the target (Required)
depends	Comma separated list of all targets that this target depends on. (Optional)
description	A short description of the target. (optional)
if	Allows the execution of a target based on the trueness of a conditional attribute. (optional)
unless	Adds the target to the dependency list of the specified Extension Point. An Extension Point is similar to a target, but it does not have any tasks. (Optional)

The **echo** task in the above example is a trivial task that prints a message. In our example, it prints the message *Hello World*.

To run the ANT build file, start command prompt and navigate to the folder where the build.xml resides, and type **ant info**. You can also type **ant** instead. Both will work, because **info** is the default target in the build file. You should see the following output:

```
C:\>ant
Buildfile: C:\build.xml
info:
    [echo] Hello World - Welcome to Apache Ant!

BUILD SUCCESSFUL
Total time: 0 seconds
C:\>
```

# 4. ANT PROPERTY TASK

ANT uses the **property** element which allows you to specify properties. This allows the properties to be changed from one build to another, or from one environment to another.

By default, ANT provides the following pre-defined properties that can be used in the build file:

Properties	Description
ant.file	The full location of the build file.
ant.version	The version of the Apache ANT installation.
basedir	The basedir of the build, as specified in the basedir attribute of the project element.
ant.java.version	The version of the JDK that is used by ANT.
ant.project.name	The name of the project, as specified in the name attribute of the project element.
ant.project.default-target	The default target of the current project.
ant.project.invoked-targets	Comma separated list of the targets that were invoked in the current project.
ant.core.lib	The full location of the ANT jar file.
ant.home	The home directory of ANT installation.
ant.library.dir	The home directory for ANT library files - typically ANT_HOME/lib folder.

ANT also makes the system properties. For Example, file.separator is available to the build file.

In addition to the above, the user can define additional properties using the **property** element. The following example shows how to define a property called **sitename**:

```
<?xml version="1.0"?>
<project name="Hello World Project" default="info">
  <property name="sitename" value="www.tutorialspoint.com"/>
  <target name="info">
    <echo>Apache Ant version is ${ant.version} - You are
    at ${sitename} </echo>
  </target>
</project>
```

Running ANT on the above build file produces the following output:

```
C:\>ant
Buildfile: C:\build.xml

info:
  [echo] Apache Ant version is Apache Ant(TM) version 1.9.4
  compiled on December 20 2014 - You are at www.tutorialspoint.com

BUILD SUCCESSFUL
Total time: 0 seconds
C:\>
```

End of ebook preview  
If you liked what you saw...  
Buy it from our store @ [\*\*https://store.tutorialspoint.com\*\*](https://store.tutorialspoint.com)