Ant is a build tool.

It aims to automate your entire build process.

Ant has an XML syntax,

Ant, you can delegate the work to the machine and add new stages to your build process.

**Ant uses XML files called build files to describe how to build a project**. Each build file describes how to build one project. Very large projects may be composed of multiple smaller projects, each with its own build file.

**In the build file:**

**1) targets and**

**2)tasks**

the high-level various goals of the build—**the targets**:

Targets represent actual outputs of the build, such as a redistributable file, or act activities, such as compiling the source or running the tests.

When declaring a target, you can declare which targets have to be built first.

When Ant builds a project, it executes targets in the order implied by their dependencies.

When Ant executes a target, it executes the tasks inside, one after the other.

actions to take to achieve each goal—**the tasks.**

**Summary of targets and tasks**

Ant reads in a build file containing a project. In the project are targets that describe different things the project can do. Inside the targets are the tasks, tasks that do the individual steps of the build. Ant executes targets in the order implied by their declared dependencies, and the tasks inside them, thereby building the application.

When the project is built, Ant determines which targets need to be executed, and in what order. Then it runs the tasks inside each target. If a task somehow fails, Ant halts the build.

as soon as you want HTML test reports, packaging, and distribution, you’ll need Ant.

***Summary for ant introduction:***

• Ant uses XML *build files* to describe what to build. Each file covers one Ant

*project*; a project is divided into *targets*; targets contain *tasks*. These tasks are the

Java classes that actually perform the construction work. Targets can depend on

other targets. Ant orders the execution so targets execute in the correct order.

• Ant is a free, open source project with broad support in the Java community.

Modern IDEs support it, as do many developer tools. It also integrates well

with modern test-centric development processes, bringing testing into the

build process.

• There are other tools that have the same function as Ant—to build software—

but Ant is the most widely used, broadly supported tool in the Java world.

• Ant is written in Java, is cross platform, integrates with all the main Java IDEs,

and has a command-line interface.

**Ant views a project: a project contains targets, each of**

**which is a set of actions—tasks—that perform part of the build. Targets can depend**

**on other targets, all of which are declared in an XML file, called a *build file*.**

**Steps for using Ant:**

1. creating the project directory :

create a directory. In the new directory, create a file called Main.java, which containing Java program:

1. verifying the tools are in place

at a command prompt type

ant -version

A good response would be something listing a recent version of Ant, version 1.7 or later:

Apache Ant version 1.7 compiled on December 13 2006

**How to install ant:**

ANT\_HOME=C:\Java\Apache\Ant

JAVA\_HOME=C:\Java\jdk

Path=C:\WINDOWS\system32;C:\WINDOWS;C:\Java\jdk\bin;C:\Java\Ant\bin

1. writing your first Ant build file

create a build.xml file as follows:

<?xml version="1.0"?>

<project name="firstbuild" default="compile" >

<target name="compile">

<javac srcdir="." />

<echo>compilation complete!</echo>

</target>

</project>

1. <project> element is always the *root* element in Ant build files,

in this case containing two attributes, name and default

1. All Ant build files must contain a single <project> element as the root element. It tells Ant the name of the project and, optionally, the default target.
2. A target represents a single stage in the build process. It can be called from the command line or it can be used internally. A build file can have many targets, each of which must have a unique name.

The <target> element is a child of <project>.

The <target> element contains two child elements: <javac> and <echo>.

1. running your first build
2. imposing structure
3. running our program

ClassPath :

Access by