Session 7 - Ant

Agenda

- Build tool: Ant
- Target and Task in Ant
- Downloading and Installing Ant
- Running Ant
- Ant’s Build File Syntax

Organizing Complex Projects

- **Use a common format for all modules of the project**
- **Compiling only new and changed files**
- **Controlling the project**
- **Reflecting on the Project's Life Cycle**

Build Tools

- **Make**
  - Script file
  - List of compiling actions
  - Cons: platform-dependent
- **Ant**
  - A Java-based build tool from the Apache project
  - Ant’s build files in XML: easy to write

Ant Targets and Tasks

- Target: a named section of the build process
- Each target contains a number of tasks
- Tasks: command-line calls as and are represented by XML elements

```
<project name="Example Application Build"
  default="default" basedir=".">
<!-- Compile the stand-alone application -
  ->
<target name="default">
<javac srcdir="./src" destdir="build"/>
<echo message="Application compiled"/>
</target>
</project>
```

Types Ant's Tasks

- ***Core tasks*: tasks that the Ant development team supports and actively develops. The team has given a commitment to look after these tasks, improve them, and correct any bugs found by Ant users.**
- ***Optional tasks*: are bundled with Ant and depend on libraries that do not belong to the Ant project (core tasks have no such dependences), and they come bundled as part of the Ant distribution, as do the libraries, so you can still use them in your projects.**
- ***Custom task*: no suitable Ant's option**
  - use Ant to run a command-line tool if one exists that does what you want to do
  - write your own task, which simply means you would write a Java class that implements the desired behavior

Installing Ant

- **Downloading a Binary Distribution:** `ant.apache.org`
- **Verifying the Download:**
  - PGP
  - MD5 and SHA1
- **Installing a Source Ant Distribution**
- **Detailed installing: see in the book and practice in lab**

Final Steps After Installation

- **Setting environment variable %ANT_HOME%**
  - In Windows:

```
Start → Settings → Control Panel →
  System
```

  - In Unix: use command

```
# ANT_HOME=/usr/java/apache-ant
```

Running Ant from the Command Line

```
ant [options] [target [target2
  [target3] ...]]
```

- Some interesting options
  - `-help, -h`   Displays the help message.
  - `-buildfile <file>` or `-file <file>` or `-f <file>`: Specifies the build file to use for this build. The value of <file> can be an absolute path or a path relative to the current directory. The default is a file called build.xml in the current directory.
- Detaited in the reference

Ant’s Build File Syntax

- The root element of every Ant build file is the <project> element
- The minimum requirements for a build file

```
<?xml version="1.0"?>
<project>
</project>
```

- Ant will always run any tasks that are encompassed by the pair of <target> </target> and child elements of <project>

Set attributes of projects

- Setting a Project Name, a Base Directory, and a Default Target

```
<?xml version="1.0"?>
<project name="Apache Ant Book Project‟
           basedir="."
           default="build-dtd">
  <target name="build-dtd">
  <antstructure output="./project.dtd"/>
  </target>
</project>
```

Target Element

- A *target* is a collection of Ant tasks that you want to run as a unit.
- Each target should represent a *discrete* and *finished* step of the build process.
- Attribute `name`:
  - Mandatory
  - Target can be called by name from the command line.
- Attribute `description`:
  - Store target’s description
  - Showed when calling Ant with with the -`projecthelp` option

Setting a Dependency between 2 targets

- Set one target always executing before running another one

```
<?xml version="1.0"?>
<project name="Apache Ant Project" basedir="."
  default="build-dtd">
  <target name="pre-dtd">
  ...
  </target>
  <target name="build-dtd" depends="pre-dtd"
           description="Create an Ant DTD">
     <antstructure output="./project.dtd"/>
  </target>
</project>
```

Working with Properties

- **Page 40**

Using Properties to Control a Build

- **Page 59**

Ant’s Types

- **Page 77**

Directory-Based Types

- **Page 77**

Pattern Sets

- **78**

Directory Sets

File Sets

Class File Sets

- **96**

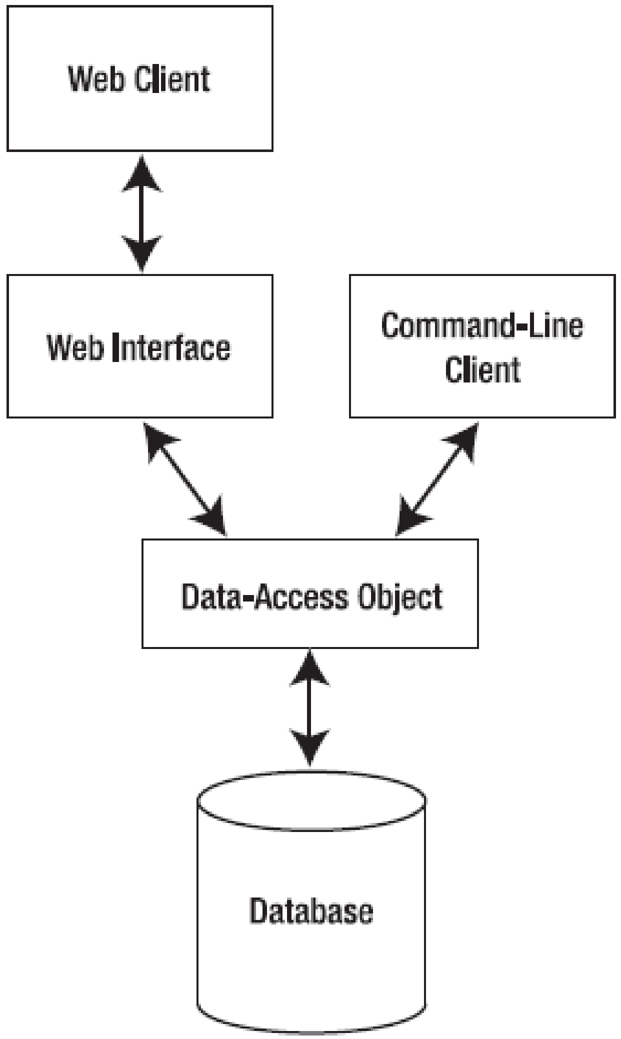
File Lists

Zip File Sets

Building a Project

- **99**

Sample Application



Modules

Compiling Java Applications with Ant

- **104**

Building the Sample Application

- 128

Deploying an Application

- **131**
- **This part only provides brief detail**

Building Documentation Bundles

Running an Application

- **169**

Testing an Application

- **187**