Build tools- ant, gradle, maven…

Maven - Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Maven is open source and it for Java based projects.

Build tool:

1.Get the required dependencies automatically from maven central repo using pom.xml-dependencies and add them to class path

Package the source code as jar/war or ear using maven plugins

It can execute automation test cases using surefire plugin in pom.xml

Pom.xml- It is main configuration file for maven project where all the required dependencies and plugins are configured.

Setup Maven :

Prerequisite:

Download java and java\_home should be set

1. Using cmd prompt

Download maven [apache-maven-3.3.9-bin.zip](http://mirrors.sonic.net/apache/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.zip) from <https://maven.apache.org/download.cgi>

Extract it to any local folder

Set env var-: add maven path till bin(C:\Program Files\apache-maven-3.3.9\bin) to existing path variable

Open a new cmd prompt window- mvn –version

1. Using IDE(eclipse,netbeans,intellij)

Install maven plugin in the ide.

In eclipse- install new software- give <http://download.eclipse.org/technology/m2e/releases> and complete the installation

Every maven Dependency has 3 main components:

groupId – package name

artifactId – project/jar/war name

version-

All dependencies can be searched from <https://mvnrepository.com/>.

.m2- is present in your local machine as maven local repository

Maven cmds:

mvn clean - cleans the prev build files/all existing files in target folder

mvn install - it gets all the required dependencies and build the project and give the specified packaged jar/war into target folder

mvn package

mvn test –

Execute automation using maven:

We need to specify testing.xml path in pom.xml using surefire plugin and view the reports under target-surefire-reports

Vcs: to maintain code at one place to be used by multiple team members at same time

centralized version control system – maintains only one central repo

distributed version control system – maintains individually –local repo and also global repo, merging and comparing the files for differences is easy.

Git : dvcs

Refer ppt for few terminologies…

Commit – move changes to local repo

Push- move changes to central repo

Fork – copying the code from some other repo to your local repo

Clone – get the forked copy into your local system or ide using clone url

Pull – fetch+merge – getting updates from central repo and merging them with local repo changes

Fetch – getting updates from central repo

Merge- merging them with local repo changes

Install git:

1. install as eclipse plugin - egit

2. cmd prompt – download git <https://git-scm.com/download/win> and execute it

3. git stash, git easy……

Refer-txt file for commands

Git and github:

Git – code changes and track of files

Github- online repository

Git commit- push to local repo

Push – which actually to central repo

Jenkins