UECS2363 SOFTWARE CONSTRUCTION AND CONFIGURATION
CHAPTER 6 : BUILD AUTOMATION

DR FARIZUWANA AKMA
farizuwana@utar.edu.my

Introduction

- Modern IDE can handle small project nicely.
- More control is needed if projects grow:
  - > 1 person
  - > several days
  - > 1 executable files
- Build tools are vital

Build Tools – An Overview

- **Automated build tools have been a part of software development for a very long time, e.g. Make and its variants.**
- **It calculates how to reach the specified goal, by executing tasks in correct order, running each tasks your goal depends on exactly once.**

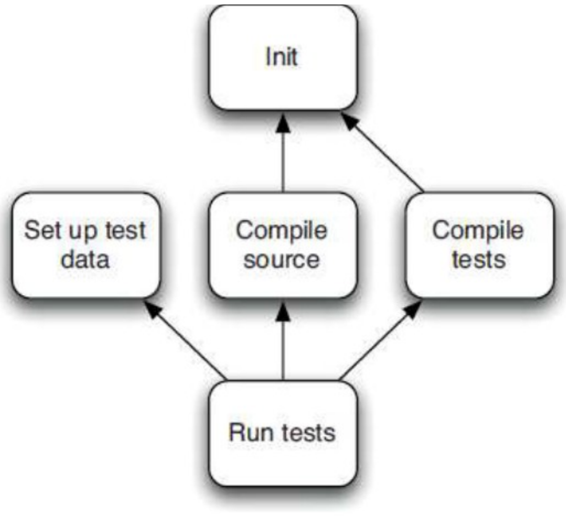


Figure 6.1 *A simple build dependency network*

Types of build tools

- **Product-Oriented Build Tools**
  - **Make**
- **Task-Oriented Build Tools**
  - **Ant, Nant, MSBuild, etc.**

Build tools

- Rake for Rails
- MSBuild for .NET
- Ant, Maven, Buildr, Gradle for Java
- SCons for C/C++

Make

- **Powerful product-oriented build tool**
- **Track dependencies within a build: build only those components that are affected by a particular change.**
- **Drawbacks:**
  - **Hard to debug when applications become more complex**
  - **Makefile problem - <space> and <tab>**
  - **Rely on shell – platform dependency**

Make

```
target: component \
        component
Tab ⭾ command ;          \
Tab ⭾ command |          \
Tab ⭾ piped-command
```

```
all: hello

hello: main.o factorial.o hello.o
        g++ main.o factorial.o hello.o -o hello

main.o: main.cpp
        g++ -c main.cpp

factorial.o: factorial.cpp
        g++ -c factorial.cpp

hello.o: hello.cpp
        g++ -c hello.cpp

clean:
        rm -rf *o hello
```

Ant

- Emerged when Java developers need more cross-platform development tool
- Fully cross-platform.
- Include a set of tasks written in Java to perform common operations, e.g. compilation, filesystem manipulation

Ant

- Can be easily extended with new tasks written in Java
- Task-oriented build tool
- Need to write build scripts in XML
- Declarative language

NAnt and MSBuild

- **When .NET was introduced, Java developers ported Ant -> NAnt**
- **Microsoft introduced minor variants of Ant -> MSBuild**

MSBuild - Overview

- Project File
  - XML format

MSBuild - Items

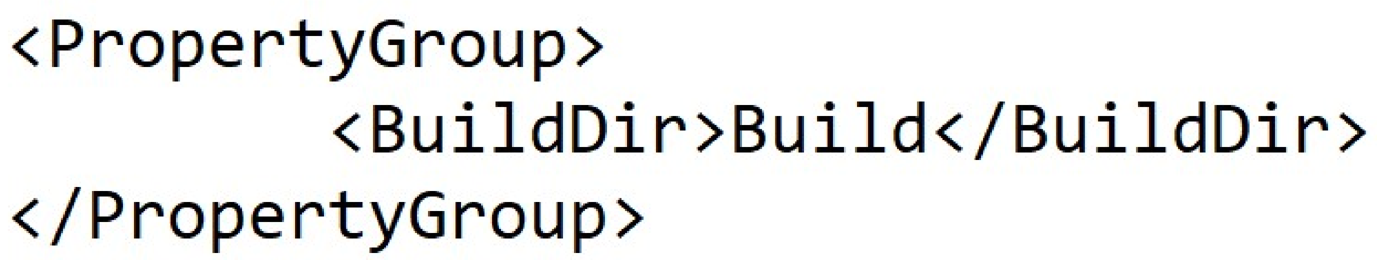
- represent inputs into the build system and are grouped into item collections.
- Can be used as parameters for tasks, which use the individual items contained in the collection to
- perform the steps of the build process.
  - E.g. item collections, named Compile:

```
<ItemGroup>
        <Compile Include = "file1.cs"/>
        <Compile Include = "file2.cs"/>
</ItemGroup>
```

- The item collection can be referenced as `@(Compile)`

MSBuild - Properties

- **Represent key/value pairs that can be used to configure builds e.g. code that create a property named `BuildDir` with a value of `Build`**



```
<PropertyGroup>
        <BuildDir>Build</BuildDir>
</PropertyGroup>
```

- **The property can be referenced as `$(BuildDir)`**

MSBuild – Items vs Properties

- Items are stored in collections, while properties contain a single scalar value.
- Items cannot be removed from item collections, while properties can have their values changed after they are defined.
- Items can contain metadata and can use the % (ItemMetadata) notation, while properties cannot.

MSBuild – Task

- **Reusable units of executable code used by MSBuild projects to perform build operations, e.g**
  - **compile input files**
  - **run an external tool**
- **The execution logic of a task is written in `<UsingTask>` element, e.g.**

```
<UsingTask TaskName="TaskName"
        AssemblyName = "AssemblyName"
        TaskFactory = "ClassName"
        Condition = "'String A'=='String B'" />
```

MSBuild – Task

- **Common build-in tasks, e.g.**
- **`MakeDir`, `Copy`, `Csc`**
- **`<MakeDir Directories="$(BuildDir)" />`**

MSBuild – Target

- **Group tasks together in a particular order and expose sections of the project file as entry points into the build process.**
- **`<Target>` element**

```
<Target Name="Compilation">
        <Csc Sources="@(Compile)" />
</Target>
```

MSBuild – Command Line

- **MSBuild.exe**
- `MyProj.proj /property:Configuration=Debug`

Maven

- **Attempts to simplify further of Ant scripts**
- **Maven will perform almost any build, deploy, test and release task with a single command, provided that your project conforms to the structure dictated by Maven**

Maven

- Drawbacks:
  - Extremely hard to build if your project doesn’t conform to Maven’s assumptions about structure and lifecycle
  - Self-updating in default configuration. May not reproduce your builds.

Other Tools

- Buildr
- Gradle

END OF LECTURE 09