Ant

Large Scale Programming, 1DL410, autumn 2009
Cons T Åhs
Automation is Everything
Automation is Everything

- Writing code and running it from inside your IDE is fine for development.
Automation is Everything

- Writing code and running it from inside your IDE is fine for development.
- It fails, however, in several circumstances:
Automate is Everything

- Writing code and running it from inside your IDE is fine for development.
- It fails, however, in several circumstances:
  - when you distribute your code as source than someone else just wants to compile and run it,
Automation is Everything

‣ Writing code and running it from inside your IDE is fine for development.

‣ It fails, however, in several circumstances:
  ‣ when you distribute your code as source than someone else just wants to compile and run it,
  ‣ when you want to build a compiled product for distribution together with additional resources needed
Automation is Everything

‣ Writing code and running it from inside your IDE is fine for development.

‣ It fails, however, in several circumstances:

‣ when you distribute your code as source than someone else just wants to compile and run it,

‣ when you want to build a compiled product for distribution together with additional resources needed

‣ when running periodic automated tests, builds, analysis etc.
Automation is Everything

- Writing code and running it from inside your IDE is fine for development.
- It fails, however, in several circumstances:
  - when you distribute your code as source than someone else just wants to compile and run it,
  - when you want to build a compiled product for distribution together with additional resources needed
  - when running periodic automated tests, builds, analysis etc.
- DRY - Don’t Repeat Yourself!
Automation is Everything

- Writing code and running it from inside your IDE is fine for development.
- It fails, however, in several circumstances:
  - when you distribute your code as source than someone else just wants to compile and run it,
  - when you want to build a compiled product for distribution together with additional resources needed
  - when running periodic automated tests, builds, analysis etc.
- DRY - Don’t Repeat Yourself!
  - This is valid for more than code.
Automation is Everything

- Writing code and running it from inside your IDE is fine for development.
- It fails, however, in several circumstances:
  - when you distribute your code as source than someone else just wants to compile and run it,
  - when you want to build a compiled product for distribution together with additional resources needed
  - when running periodic automated tests, builds, analysis etc.
- DRY - Don’t Repeat Yourself!
  - This is valid for more than code.
  - It is also valid for everything that surrounds a project, such as running tests and builds etc.
Automation is Everything

‣ Writing code and running it from inside your IDE is fine for development.

‣ It fails, however, in several circumstances:
  ‣ when you distribute your code as source than someone else just wants to compile and run it,
  ‣ when you want to build a compiled product for distribution together with additional resources needed
  ‣ when running periodic automated tests, builds, analysis etc.

‣ DRY - Don’t Repeat Yourself!
  ‣ This is valid for more than code.
  ‣ It is also valid for everything that surrounds a project, such as running tests and builds etc.
  ‣ Remember, we want to tests to be automated, i.e., not containing manual procedures.
Automation is Everything

- Writing code and running it from inside your IDE is fine for development.
- It fails, however, in several circumstances:
  - when you distribute your code as source than someone else just wants to compile and run it,
  - when you want to build a compiled product for distribution together with additional resources needed
  - when running periodic automated tests, builds, analysis etc.
- DRY - Don’t Repeat Yourself!
  - This is valid for more than code.
  - It is also valid for everything that surrounds a project, such as running tests and builds etc.
  - Remember, we want to tests to be automated, i.e., not containing manual procedures.
  - This goes for compiling, building, analysis, distribution as well.
Automation is Everything

‣ Writing code and running it from inside your IDE is fine for development.
‣ It fails, however, in several circumstances:
  ‣ when you distribute your code as source than someone else just wants to compile and run it,
  ‣ when you want to build a compiled product for distribution together with additional resources needed
  ‣ when running periodic automated tests, builds, analysis etc.
‣ DRY - Don’t Repeat Yourself!
  ‣ This is valid for more than code.
  ‣ It is also valid for everything that surrounds a project, such as running tests and builds etc.
  ‣ Remember, we want to tests to be automated, i.e., not containing manual procedures.
  ‣ This goes for compiling, building, analysis, distribution as well.
‣ Ant comes to the rescue!
Automated is Everything

- Writing code and running it from inside your IDE is fine for development.
- It fails, however, in several circumstances:
  - when you distribute your code as source than someone else just wants to compile and run it,
  - when you want to build a compiled product for distribution together with additional resources needed
  - when running periodic automated tests, builds, analysis etc.
- DRY - Don’t Repeat Yourself!
  - This is valid for more than code.
  - It is also valid for everything that surrounds a project, such as running tests and builds etc.
  - Remember, we want to tests to be automated, i.e., not containing manual procedures.
  - This goes for compiling, building, analysis, distribution as well.
- Ant comes to the rescue!
  - A portable (written in Java) build tool for automating project related tasks.
Automation is Everything

‣ Writing code and running it from inside your IDE is fine for development.
‣ It fails, however, in several circumstances:
  ‣ when you distribute your code as source than someone else just wants to compile and run it,
  ‣ when you want to build a compiled product for distribution together with additional resources needed
  ‣ when running periodic automated tests, builds, analysis etc.
‣ DRY - Don’t Repeat Yourself!
  ‣ This is valid for more than code.
  ‣ It is also valid for everything that surrounds a project, such as running tests and builds etc.
  ‣ Remember, we want to tests to be automated, i.e., not containing manual procedures.
  ‣ This goes for compiling, building, analysis, distribution as well.
‣ Ant comes to the rescue!
  ‣ A portable (written in Java) build tool for automating project related tasks.
  ‣ ant has a precursor in make (not quite as portable, though)
Automation is Everything
Automation is Everything

- Typical things you want to do with a build tool:
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
  - create distributions
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
  - create distributions
    - source code
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
  - create distributions
    - source code
    - compiled products with relevant resources
Automated is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
  - create distributions
    - source code
    - compiled products with relevant resources
  - run analysis tools on source code
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
  - create distributions
    - source code
    - compiled products with relevant resources
  - run analysis tools on source code

- Ant has built in support for a lot of standard build related tasks and can be extended with new tasks.
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
  - create distributions
    - source code
    - compiled products with relevant resources
  - run analysis tools on source code
- Ant has built in support for a lot of standard build related tasks and can be extended with new tasks.
  - It is really not limited to Java projects
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
  - create distributions
    - source code
    - compiled products with relevant resources
  - run analysis tools on source code
- Ant has built in support for a lot of standard build related tasks and can be extended with new tasks.
  - It is really not limited to Java projects
- Ant reads a description file, typically `build.xml`, and fulfils (executes) one or several tasks.
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
  - create distributions
    - source code
    - compiled products with relevant resources
  - run analysis tools on source code
- Ant has built in support for a lot of standard build related tasks and can be extended with new tasks.
  - It is really not limited to Java projects
- Ant reads a description file, typically `build.xml`, and fulfils (executes) one or several tasks.
  - A task can depend on other tasks, which means that the other tasks must be fulfilled first. They are prerequisites/dependants of the original task.
Automation is Everything

- Typical things you want to do with a build tool:
  - compile all source code
  - run tests
  - create distributions
    - source code
    - compiled products with relevant resources
  - run analysis tools on source code

- Ant has built in support for a lot of standard build related tasks and can be extended with new tasks.
  - It is really not limited to Java projects

- Ant reads a description file, typically build.xml, and fulfils (executes) one or several tasks.
  - A task can depend on other tasks, which means that the other tasks must be fulfilled first. They are prerequisites/dependants of the original task.
  - Task execution order is determined by a topological sorting based on the dependency graph.
Ant input: build.xml
Ant input: build.xml

- The input to ant is borderline readable as it is a file containing an XML document.
The input to ant is borderline readable as it is a file containing an XML document.

The root element is `project`, with the attribute `default` specifying the default task.
Ant input: build.xml

- The input to ant is borderline readable as it is a file containing an XML document.

- The root element is `project`, with the attribute `default` specifying the default task.

- Each task is a sub element `target`, with the attribute `name` specifying the name of the task/target.
Ant input: build.xml

- The input to ant is borderline readable as it is a file containing an XML document.
- The root element is `project`, with the attribute `default` specifying the default task.
- Each task is a sub element `target`, with the attribute `name` specifying the name of the task/target.
- The sub elements of `target` are the sub tasks that are to be executed, e.g., compiling source code etc.
Ant input: build.xml

- The input to ant is borderline readable as it is a file containing an XML document.

- The root element is `project`, with the attribute `default` specifying the default task.

- Each task is a sub element `target`, with the attribute `name` specifying the name of the task/target.

- The sub elements of `target` are the sub tasks that are to be executed, e.g., compiling source code etc.

```xml
<project default="compile">  
  <target name="compile">  
    <javac srcdir="src">  
    </javac>  
  </target>  
  <target name="compile-debug">  
    <javac srcdir="src" destdir="debug" debug="true"/>  
  </target>  
</project>
```
Ant input
Ant input

- It is also possible to set properties in the input, e.g., to give names to directories in just one place and then use the name.
Ant input

- It is also possible to set properties in the input, e.g., to give names to directories in just one place and then use the name.
  - We can thus use the same care when writing the `build.xml` as when writing all code.
Ant input

- It is also possible to set properties in the input, e.g., to give names to directories in just one place and then use the name.
  - We can thus use the same care when writing the `build.xml` as when writing all code.
- Properties are references using `${property}`
Ant input

- It is also possible to set properties in the input, e.g., to give names to directories in just one place and then use the name.
  - We can thus use the same care when writing the `build.xml` as when writing all code.
- Properties are references using `${property}`
- Properties can be set in a number of ways
Ant input

- It is also possible to set properties in the input, e.g., to give names to directories in just one place and then use the name.
  - We can thus use the same care when writing the `build.xml` as when writing all code.
- Properties are references using `${property}`
- Properties can be set in a number of ways
  - `<property name="movies" value="${outdir}/movie"/>`
Ant input

- It is also possible to set properties in the input, e.g., to give names to directories in just one place and then use the name.
  - We can thus use the same care when writing the build.xml as when writing all code.
- Properties are references using ${property}
- Properties can be set in a number of ways
  - `<property name="movies" value="${outdir}/movie"/>
  - `<property file="project.properties"/> - read properties (in another format) from a file.
Ant
Ant

- Ant can help with all automated tasks you need in conjunction with your project.
Ant

- Ant can help with all automated tasks you need in conjunction with your project.

- If you find that you are repeating manual steps, find a way to include these as a new task in your *build.xml*. The time spent constructing the task will save you time later (when you really need it).
Ant

- Ant can help with all automated tasks you need in conjunction with your project.
- If you find that you are repeating manual steps, find a way to include these as a new task in your `build.xml`. The time spent constructing the task will save you time later (when you really need it).
- You can find a lot of information on Ant on the net. Good places to start are...
Ant

- Ant can help with all automated tasks you need in conjunction with your project.
- If you find that you are repeating manual steps, find a way to include these as a new task in your `build.xml`. The time spent constructing the task will save you time later (when you really need it).
- You can find a lot of information on Ant on the net. Good places to start are
  - http://ant.apache.org/
Ant

- Ant can help with all automated tasks you need in conjunction with your project.
- If you find that you are repeating manual steps, find a way to include these as a new task in your `build.xml`. The time spent constructing the task will save you time later (when you really need it).
- You can find a lot of information on Ant on the net. Good places to start are
Ant

- Ant can help with all automated tasks you need in conjunction with your project.
- If you find that you are repeating manual steps, find a way to include these as a new task in your build.xml. The time spent constructing the task will save you time later (when you really need it).
- You can find a lot of information on Ant on the net. Good places to start are
  - http://ant.apache.org/
- Spend time learning at least the basics of Ant and expand your knowledge when needed.