Automated Builds

Introduction
A typical working day
A typical working day

Re-draw

Figure 8
A typical working day

Re-draw Figure 8

Recalculate data
A typical working day

- Re-draw
  - Figure 8

- Recalculate
  - data

- Recompile
  - stats program
A typical working day

- Re-draw
- Figure 8
- Recalculate data
- Recompile stats program
- Update Java

Introduction
A typical working day

- Re-draw
  - Figure 8
- Recalculate data
- Recompile stats program
- Re-install graph tool
- Update Java

Automated Builds

Introduction
A typical working day

- Re-draw Figure 8
- Recalculate data
- Recompile stats program
- Re-install graph tool
- Update Java

Introduction
A typical working day

- Recalculate data
- Recompile stats program
- Re-install graph tool
- Re-draw Figure 8
- Update Java
- Free up disk space

Automated Builds
A typical working day

- Re-draw
  - Figure 8
- Recalculate
  - data
- Re-compile
  - stats program
- Re-install
  - graph tool
- Update
  - Java
- Free up
  - disk space

Automated Builds

Introduction
A typical working day

Re-draw Figure 8

Re-calculate data

Re-compile stats program

Re-install graph tool

Update Java

Free up disk space

...shave the yak...
Automated Builds

Introduction

- Re-draw Figure 8
- Recalculate data
- Re-install graph tool
- Re-compile stats program
- Update Java
- Free up disk space
Automated Builds

**task**
- Re-draw
- Figure 8
- Recalculate
- data
- Re-compute
- stats program
- Re-install
- graph tool
- Update
- Java
- Free up
- disk space

**dependencies**
Automated Builds

Re-draw

Re-install

Recalculate
data

Recompile
stats program

Update
Java

Free up
disk space

task

dependencies

Re-draw

Figure 8

Recalculate
data

Re-install

graph tool

Recompile
stats program

Update
Java

Free up
disk space

Automated Builds

Introduction
This pattern arises frequently
This pattern arises frequently

New data collected?
Recalculate statistics
This pattern arises frequently

New data collected?
Recalculate statistics

Source files changed?
Recompile program
This pattern arises frequently

New data collected?
Recalculate statistics

Source files changed?
Recompile program

New content written?
Update web site
Hard or impossible to keep track of:
Hard or impossible to keep track of:

- what depends on what
Hard or impossible to keep track of:
- what depends on what
- what's up-to-date and what isn't
Hard or impossible to keep track of:
- what depends on what
- what's up-to-date and what isn't

"Anything worth repeating is worth automating."
Hard or impossible to keep track of:
- what depends on what
- what's up-to-date and what isn't
"Anything worth repeating is worth automating."

So use a *build manager* to automate the process
Hard or impossible to keep track of:
– what depends on what
– what's up-to-date and what isn't
"Anything worth repeating is worth automating."
So use a build manager to automate the process
Describe dependencies in a build file
Hard or impossible to keep track of:
- what depends on what
- what's up-to-date and what isn't

"Anything worth repeating is worth automating."
So use a *build manager* to automate the process
Describe dependencies in a *build file*
Along with commands used to update things
Hard or impossible to keep track of:
- what depends on what
- what's up-to-date and what isn't

"Anything worth repeating is worth automating."
So use a *build manager* to automate the process
Describe dependencies in a *build file*
Along with commands used to update things

Build manager does the rest
Most widely used build manager is Make
Most widely used build manager is Make

Note: "most widely used", not "most popular"
Most widely used build manager is Make
Note: "most widely used", not "most popular"
Invented by a student intern at Bell Labs in 1975
Most widely used build manager is Make
Note: "most widely used", not "most popular"
Invented by a student intern at Bell Labs in 1975
Has grown into a little programming language
Most widely used build manager is Make
Note: "most widely used", not "most popular"
Invented by a student intern at Bell Labs in 1975
Has grown into a little programming language
A very cryptic little language, without a debugger...
Most widely used build manager is Make
Note: "most widely used", not "most popular"
Invented by a student intern at Bell Labs in 1975
Has grown into a little programming language
A very cryptic little language, without a debugger...
...that requires an understanding of the Unix shell
GNU Make is fast, free, and well-documented
GNU Make is fast, free, and well-documented

And many other tools know how to work with it
GNU Make is fast, free, and well-documented
And many other tools know how to work with it
Look at basics and a few advanced features
GNU Make is fast, free, and well-documented. And many other tools know how to work with it. Look at basics and a few advanced features.

Companion lecture explores SCons.
GNU Make is fast, free, and well-documented. And many other tools know how to work with it. Look at basics and a few advanced features.

Companion lecture explores SCons.

Java users should look at Ant.