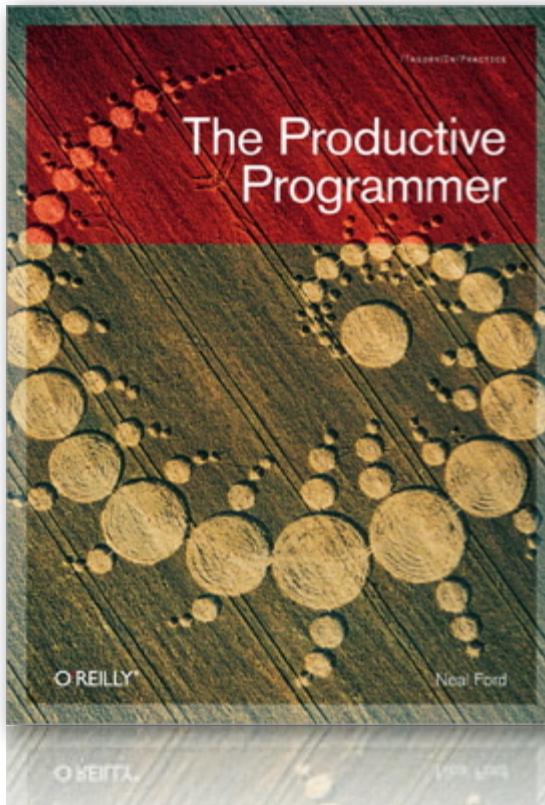


# the productive programmer: mechanics



NEAL FORD thoughtworker / meme wrangler

**ThoughtWorks**

14 Wall St, Suite 2019, New York, NY 10005

nford@thoughtworks.com  
[www.nealford.com](http://www.nealford.com)  
[www.thoughtworks.com](http://www.thoughtworks.com)  
[memeagora.blogspot.com](http://memeagora.blogspot.com)

мемеагора.blogspot.com  
[www.thoughtworks.com](http://www.thoughtworks.com)

nealford.com

 nealford.com

 Art of Java Web Development

 The DSW Group

 Manning Publications

 ThoughtWorks

**Neal Ford**  
**ThoughtWorker / Meme Wrangler**

Welcome to the web site of Neal Ford. The purpose of this site is twofold. First, it is an informational site about my professional life, including appearances, articles, presentations, etc. For this type of information, consult the news page (this page) and the [About Me](#) pages.

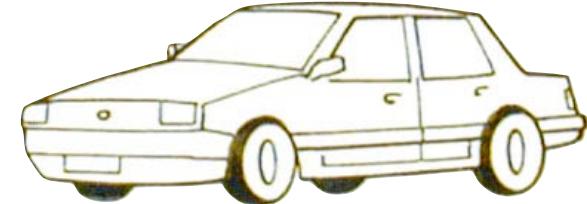
The second purpose for this site is to serve as a forum for the things I enjoy and want to share with the rest of the world. This includes (but is not limited to) reading (Book Club), Triathlon, and Music. This material is highly individualized and all mine!

Please feel free to browse around. I hope you enjoy what you find.

**Upcoming Conferences**

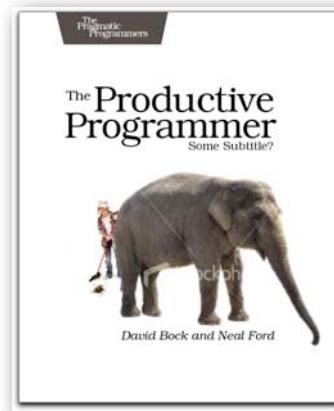
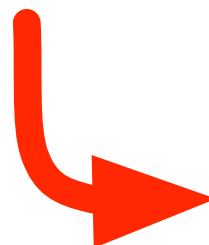


# where did this topic come from?

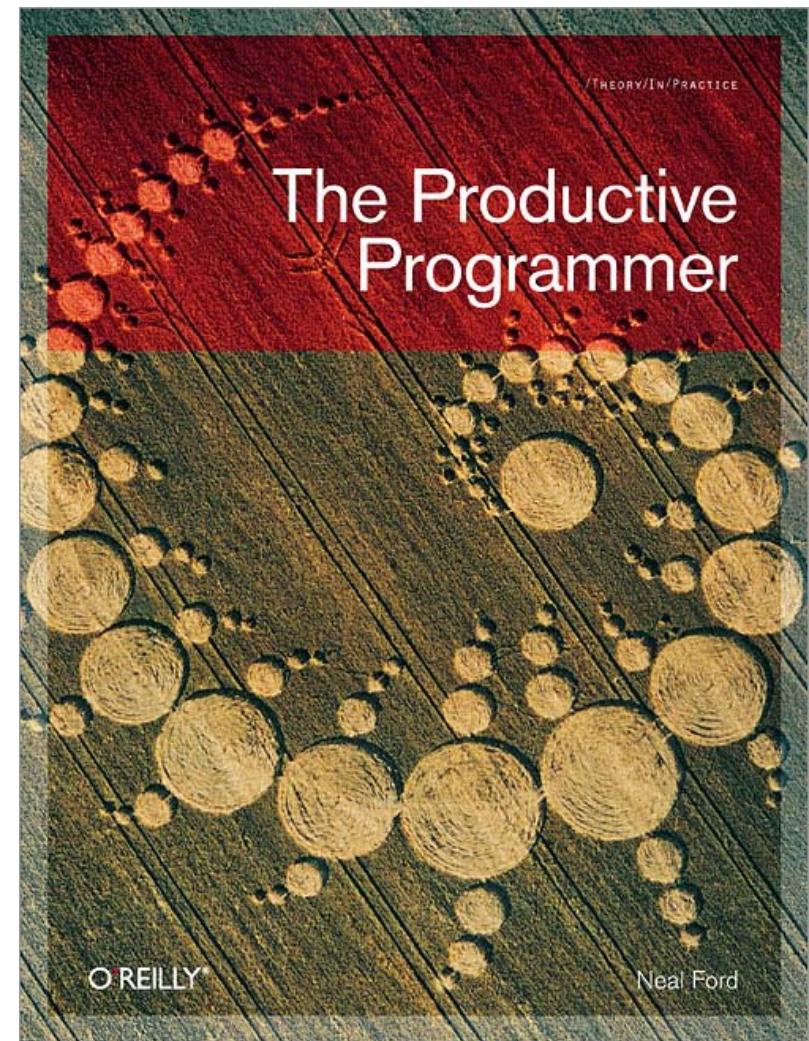
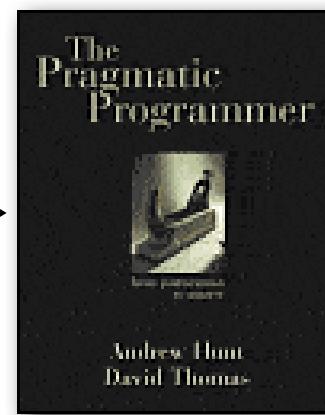
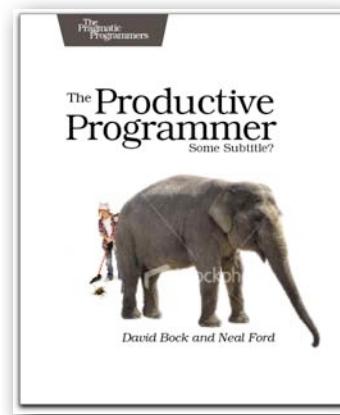


# where did this topic come from?

```
Default screen shots
[nealford] ~ ]=> ls
2print
Applications
Archive
Desktop
Documents
Downloads
Images
Library
Mingle
Movies
Music
Neal_Ford-LOP_Shifting_Paradigms-slides.pdf
NetBeansProjects
Notes702Mac.dmg
PDFs
PartitionMagic805AllWin_English.ZIP
Pictures
Public
Sites
[nealford] ~ ]=> cd ~/work/nealford/
[nealford] ~/work/nealford ]=> svn st | grep '^?*' | tr '?*' ' ' | sed 's/[ ]*//'
| sed 's/[ ]/\ /g' | xargs svn add
```



# where did this topic come from?



# 2 parts

mechanics

focus, acceleration, automation, canonicality

practices

10 ways to improve your code

# what i cover

acceleration

making the computer do stuff faster

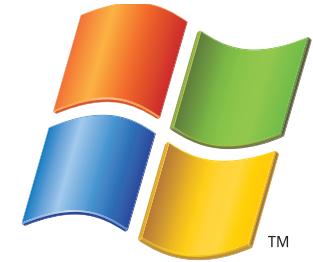
automation

making the computer do more stuff for you

# acceleration



# launching



how many columns do you have in your  
*programs* listing?

finding applications == needle in a haystack

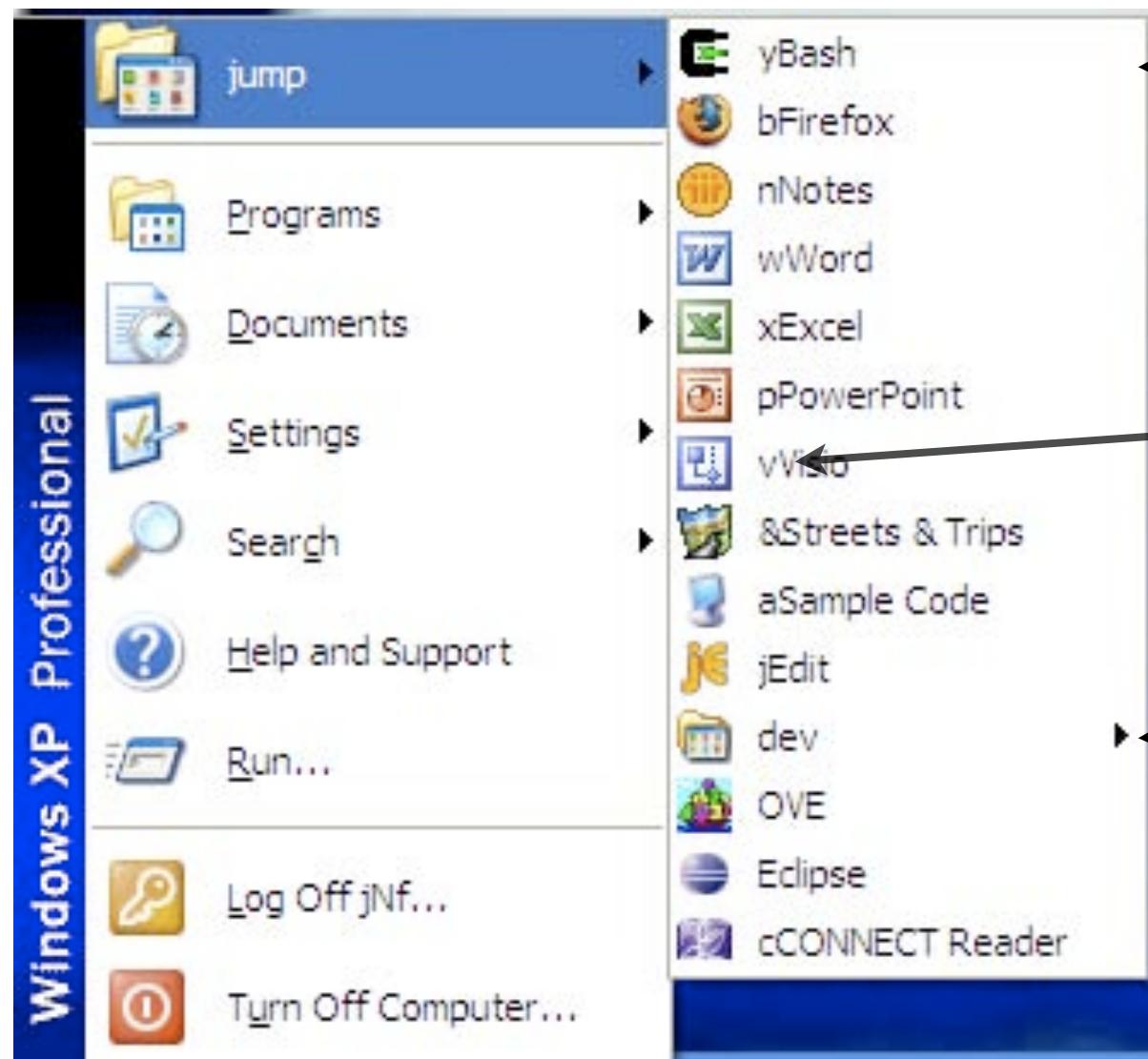
developers have lots of occasional applications

typing is faster than navigation



# create a launch folder

regular  
folder

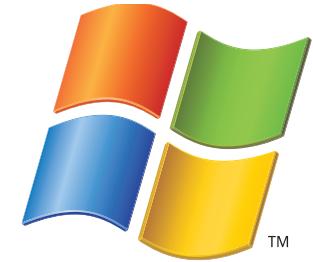


← stuff you  
run all the  
time

unique 1st  
letters

→ groups if  
necessary

# launchers



launchy

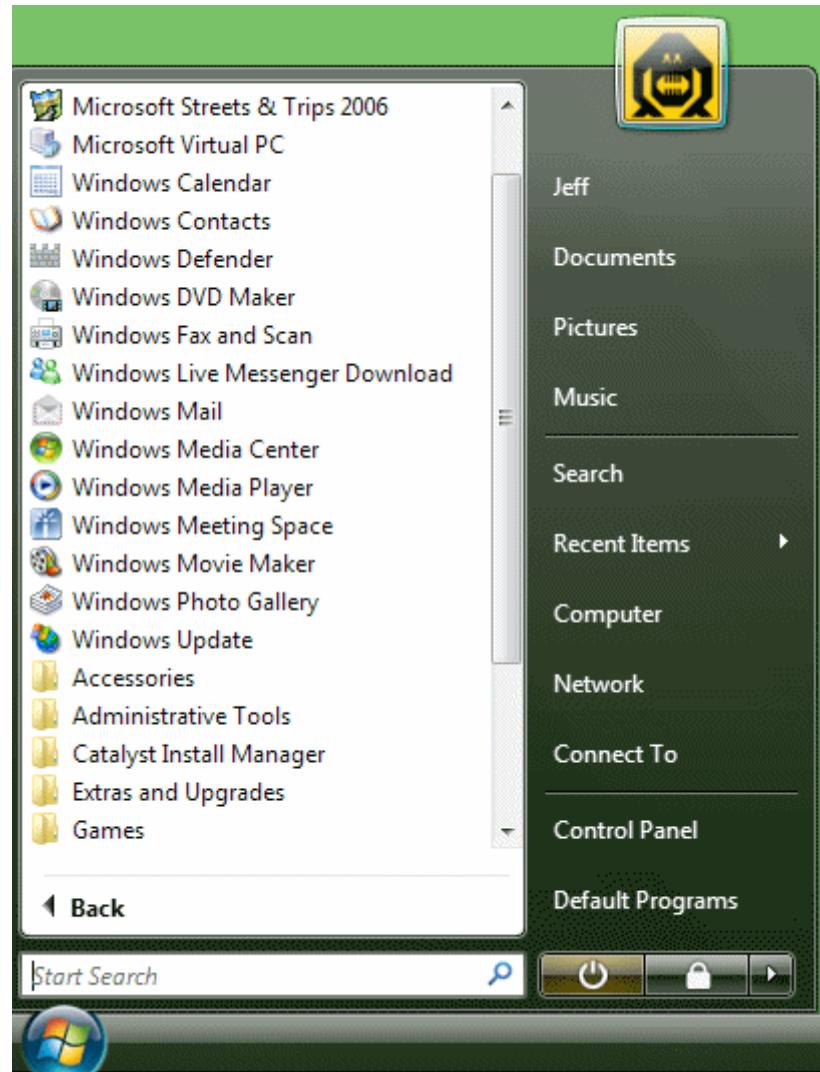
colibri

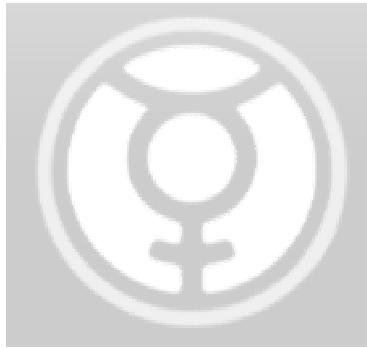


enso



# vista launcher





# quicksilver



graphical command line

finder replacement

clipboard history utility

iTunes driver

subversion client



# spotlight



search utility...

...search for applications

slow-ish in tiger

fast in leopard

# gnome



alt-f2

# o/s accelerators

windows explorer address bar (**alt-d**)

finder (**apple-shift-g**)

firefox

/ searching

number-fox plugin

# leopard smart help



Screenshot of the TextMate application interface demonstrating the "Wrap Column" feature.

The menu bar shows: View, Text, Navigation, Bundles, Window, Help.

The "Help" menu is open, showing a search field with "wrap" and a list of results:

- Gutter > Soft Wrap Indicators
- Soft Wrap
- Wrap Column** (highlighted with a blue arrow)
- Groovy > Wrap Selection in Try/...
- HTML > Wrap Selection in Open/...
- HTML > Wrap Each Selected Line ...
- HTML > Wrap in <?= ... ?>
- Hyperlink Helper > Wrap Word a...
- LaTeX > Wrap Selection In
- PML > Wrap It
- Snippets
- Expert Preferences
- TextMate Manual
- Bundles
- Key Bindings
- Show All Results

The main editor window displays some Java code:`PersonDemo() {  
 Job job = new Job("Safety Engineer");  
 Person homer = new Person("Homer", "Simpson");  
 homer.changeJobPosition("Janitor");  
  
 void changeJobPositionToString(String position) {  
 job.setJobTitle(position);  
 }  
}`

# clipboards

why do operating systems have only 1 clipboard with 1 entry????

clcl



iClip (\$\$)

jump cut



**context switching eats time**



# there and back

**pushd** pushes a directory on the stack

**popd** pops it back off

```
C:\temp>pushd \MyDocuments\Documents\dev\haskell
```

```
C:\MyDocuments\Documents\dev\haskell>dir
02/13/2006  12:12 AM    <DIR>      .
02/13/2006  12:12 AM    <DIR>      ..
02/13/2006  12:12 AM    <DIR>      nfjs_functionallangs_haskell
                  0 File(s)            0 bytes
                  3 Dir(s)  11,743,178,752 bytes free
```

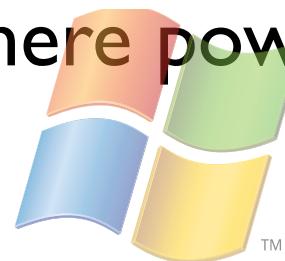
```
C:\MyDocuments\Documents\dev\haskell>popd
```

# command prompts

graphical explorers better for some things....

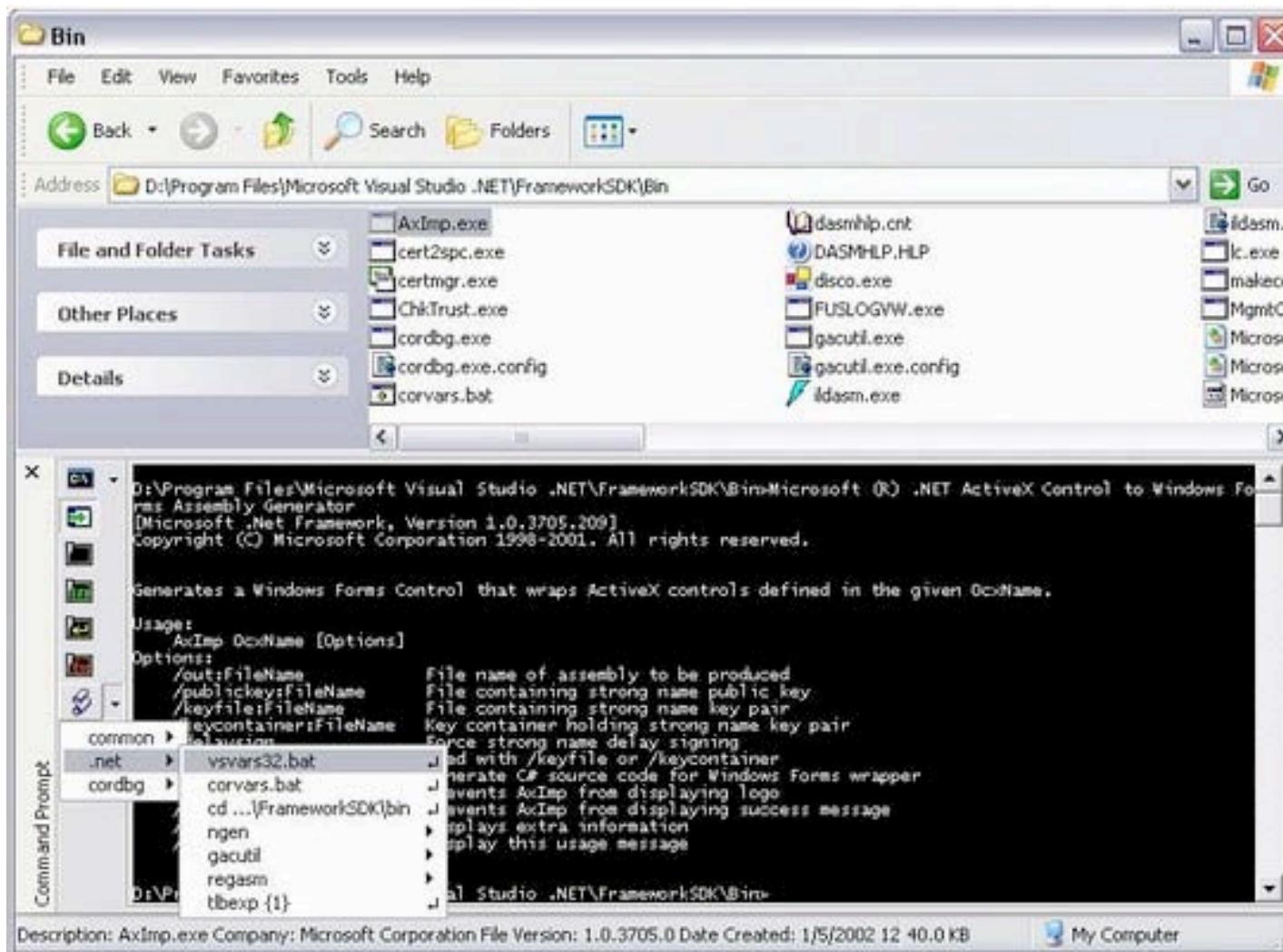
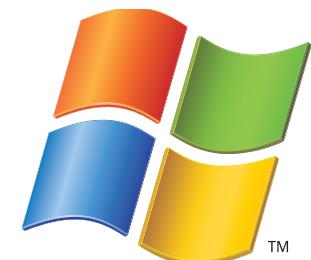
...command line better for others

command prompt here power toy



bash here (cygwin)

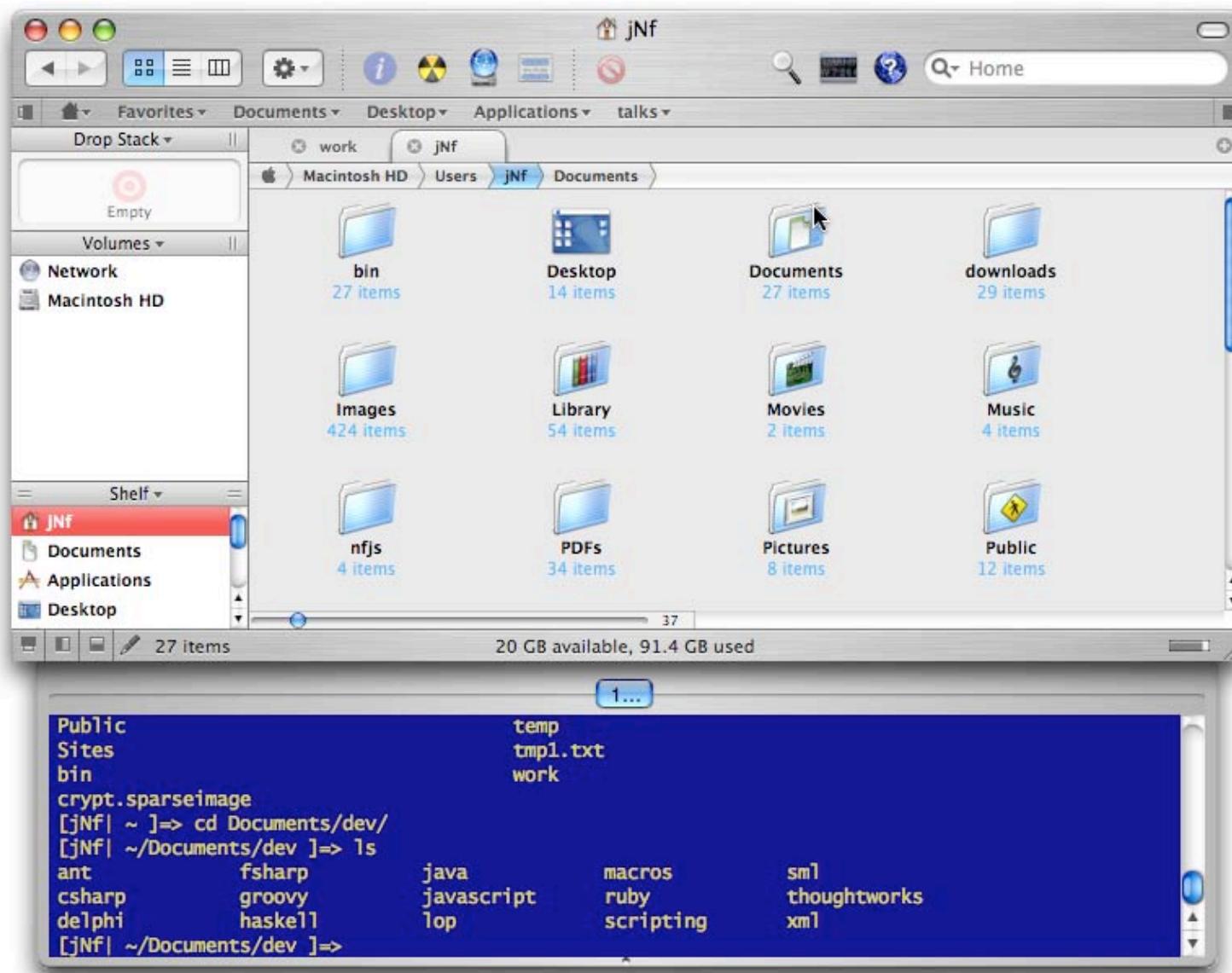
# command prompt explorer bar



Description: AxImp.exe Company: Microsoft Corporation File Version: 1.0.3705.0 Date Created: 1/5/2002 12:40:00 KB

My Computer

# path finder



how many of you have  
written an application for  
heads-down data entry  
personnel?

**when coding, always prefer  
keyboard to mouse**



# learning shortcuts

make yourself use the shortcut even if you've gotten there another way

have someone/something pester you about it

pair programmer

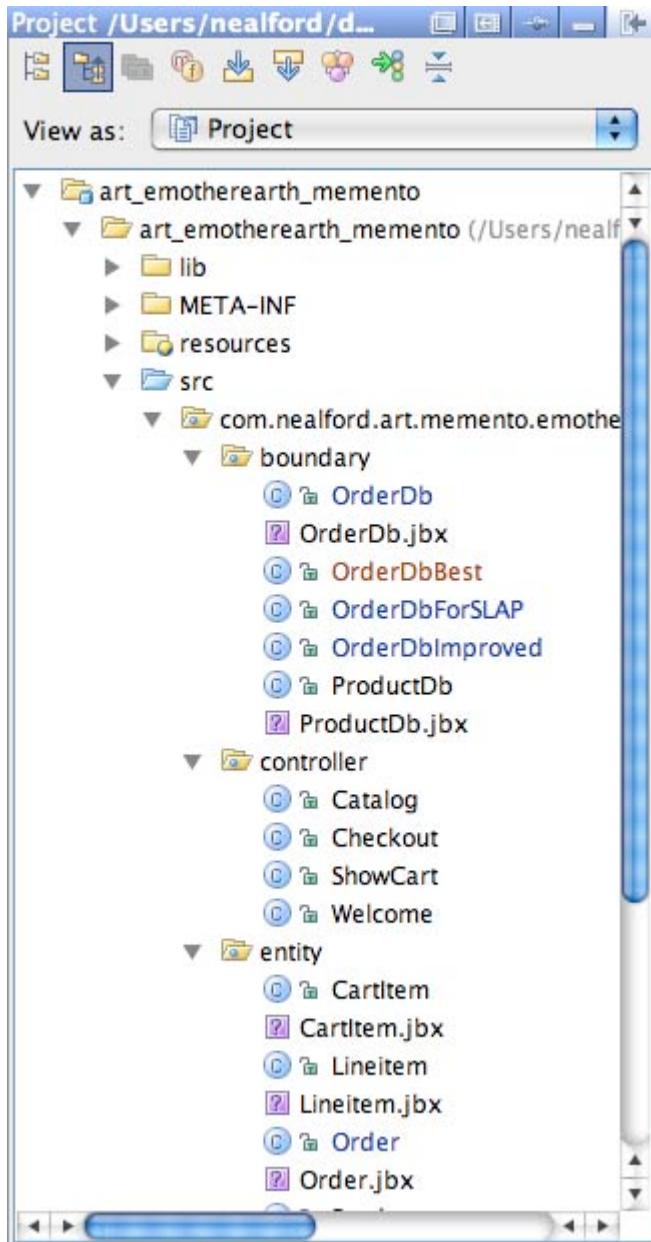
key promoter plug-in for intellij

key prompter for eclipse

# learning shortcuts

repeat them to yourself

flash cards



all our  
hierarchies  
are too deep:  
  
file system  
packages

```
private static final String TO_RETRIEVE_USER_KEY =
    "SELECT ID FROM USERS WHERE NAME = ?";
private static final String SQL_INSERT_LINEITEM =
    "INSERT INTO LINEITEMS (ORDER_KEY, ITEM_ID, QUANTITY) " +
        "VALUES(?, ?, ?)";
private static final String SQL_INSERT_ORDER =
    "INSERT INTO ORDERS (USER_KEY, CC_TYPE, CC_NUM, CC_EXP)" +
        "+ VALUES (?, ?, ?, ?, ?)";
private DBPool dbPool;

public void addOrderFrom(ShoppingCart cart, String userName,
                         Order order) throws SQLEXception {
    Map db = setupDataInfrastructure();
    try {
        int userKey = userKeyBasedOn(userName, db);
        add(order, userKey, db);
        addLineItemsFrom(cart,
                         order.getOrderKey(), db);
        completeTransaction(db);
    } catch (SQLEXception sqlx) {
        rollbackTransactionFor(db);
        throw sqlx;
    } finally {
        cleanUp(db);
    }
}
```

goto class

```
private static final String TO_RETRIEVE_USER_KEY =
    "SELECT ID FROM USERS WHERE NAME = ?";
private static final String SQL_INSERT_LINEITEM =
    "INSERT INTO LINEITEMS (ORDER_KEY, ITEM_ID, QUANTITY) " +
        "VALUES(?, ?, ?)";
private static final String SQL_INSERT_ORDER =
    "INSERT INTO ORDERS (USER_KEY, CC_TYPE, CC_NUM, CC_EXP)" +
        "+ VALUES (?, ?, ?, ?, ?)";
private DBPool dbPool;
```

```
public void addOrderFrom(ShoppingCart cart, String userName,
                         Order order) throws SQLException {
    Map db = setupDataInfrastructure();
    try {
        int userKey = userKeyBasedOn(userName, db);
        add(order, userKey, db);
        addLineItemsFrom(cart,
                         order.getOrderKey(), db);
        completeTransaction(db);
    } catch (SOLEException sqlx) {
        rollbackTransactionFor(db);
        throw sqlx;
    } finally {
        cleanUp(db);
    }
}
```

goto class: pattern of capital letters



art\_emotherearth\_memento src com nealford art memento emotherearth util ShoppingCart

OrderDbBest.java ShoppingCart.java

```
public void restoreFromBookmark(ShoppingCartMemento memento) {
    this.itemList = memento.restoreMemento();
}

public class ShoppingCartMemento {
    private List itemList;

    public List restoreMemento() {
        return itemList;
    }

    public void saveMemento() {
        List mementoList = ShoppingCart.this.itemList;
        itemList = new ArrayList(mementoList.size());
        Iterator i = mementoList.iterator();
        while (i.hasNext())
            itemList.add(i.next());
    }
}
```

introduce variable

6: TODO

CheckStyle

Regex

9: Changes

56:36

Insert

Default

55M of 69M

```
public void restoreFromBookmark(ShoppingCartMemento memento) {
    this.itemList = memento.restoreMemento();
}

public class ShoppingCartMemento {
    private List itemList;

    public List restoreMemento() {
        return itemList;
    }

    public void saveMemento() {
        List mementoList = ShoppingCart.this.itemList;
        itemList = new ArrayList(mementoList.size());
        Iterator i = mementoList.iterator();
        while (i.hasNext())
            itemList.add(i.next());
    }
}
```

# introduce variable redux

```
public void addOrder(final ShoppingCart cart, String userName,
                     Order order) throws SQLException {
    connection = null;
    stmt = null;
    try {
        connection = dbPool.getConnection();
        insertOrder(getUserKey(userName), order);
        int orderKey = getGeneratedOrderKey();
        insertLineItems(cart, orderKey);
        commitOrder(order, orderKey);
    } catch (SQLException sqlx) {
        connection.rollback();
        throw sqlx;
    } finally {
        try {
            dbPool.release(connection);
            if (stmt != null) {
                stmt.close();
            }
        } catch (SQLException ignored) {
        }
    }
}

private void insertLineItems(final ShoppingCart cart, int orderKey) th
```

escalating  
selection

```
public void addOrder(final ShoppingCart cart, String userName,
                     Order order) throws SQLException {
    connection = null;
    stmt = null;
    try {
        connection = dbPool.getConnection();
        insertOrder(getUserKey(userName), order);
        int orderKey = getGeneratedOrderKey();
        insertLineItems(cart, orderKey);
        commitOrder(order, orderKey);
    } catch (SQLException sqlx) {
        connection.rollback();
        throw sqlx;
    } finally {
        try {
            dbPool.release(connection);
            if (stmt != null) {
                stmt.close();
            }
        } catch (SQLException ignored) {
        }
    }
}
```

goto symbol

# goto symbol

```
private void insertLineItems(final ShoppingCart cart, int orderKey) {
    Iterator<Item> it = cart.getItemList().iterator();
```

# some choice shortcuts

	<i>intelliJ</i>	<i>eclipse</i>
goto class	ctrl-n	ctrl-shift-t
introduce variable	ctrl-alt-v	alt-shift-l
escalating selection	ctrl-w	alt-shift-up
recently edited files	ctrl-e	n/a (ctrl-e)
symbol list	alt-ctrl-shift-n	ctrl-o
incremental search	alt-f3	ctrl-j

# live templates

all major ide's and coding text editors

parameter substitution, default values, repeating  
values

learn the language of your template engine

velocity in intellij

bash for textmate/e editor

**every time you type  
something for the 3rd time,  
templatize it**



# key macro tools

live templates at the o/s level

auto-hot key



textexpander



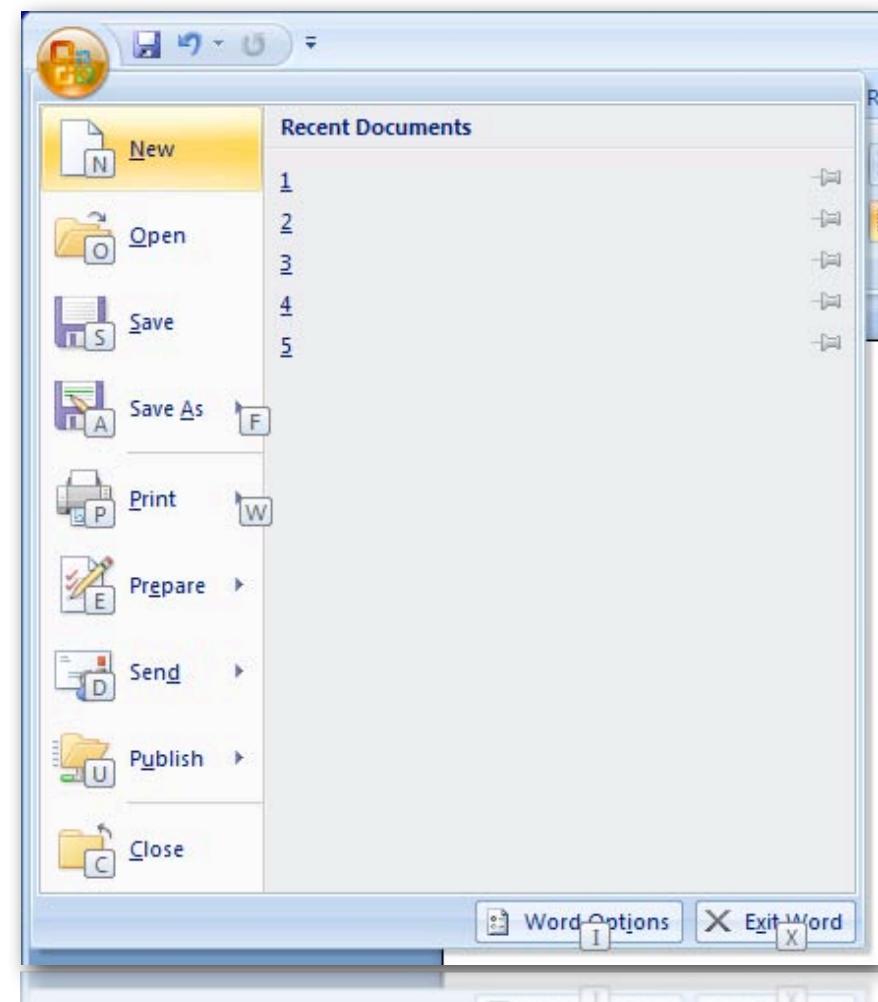
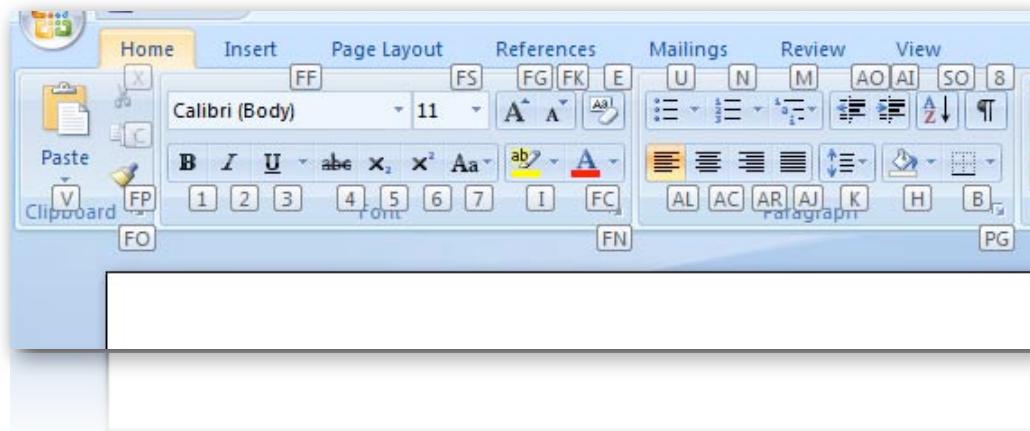
typinator

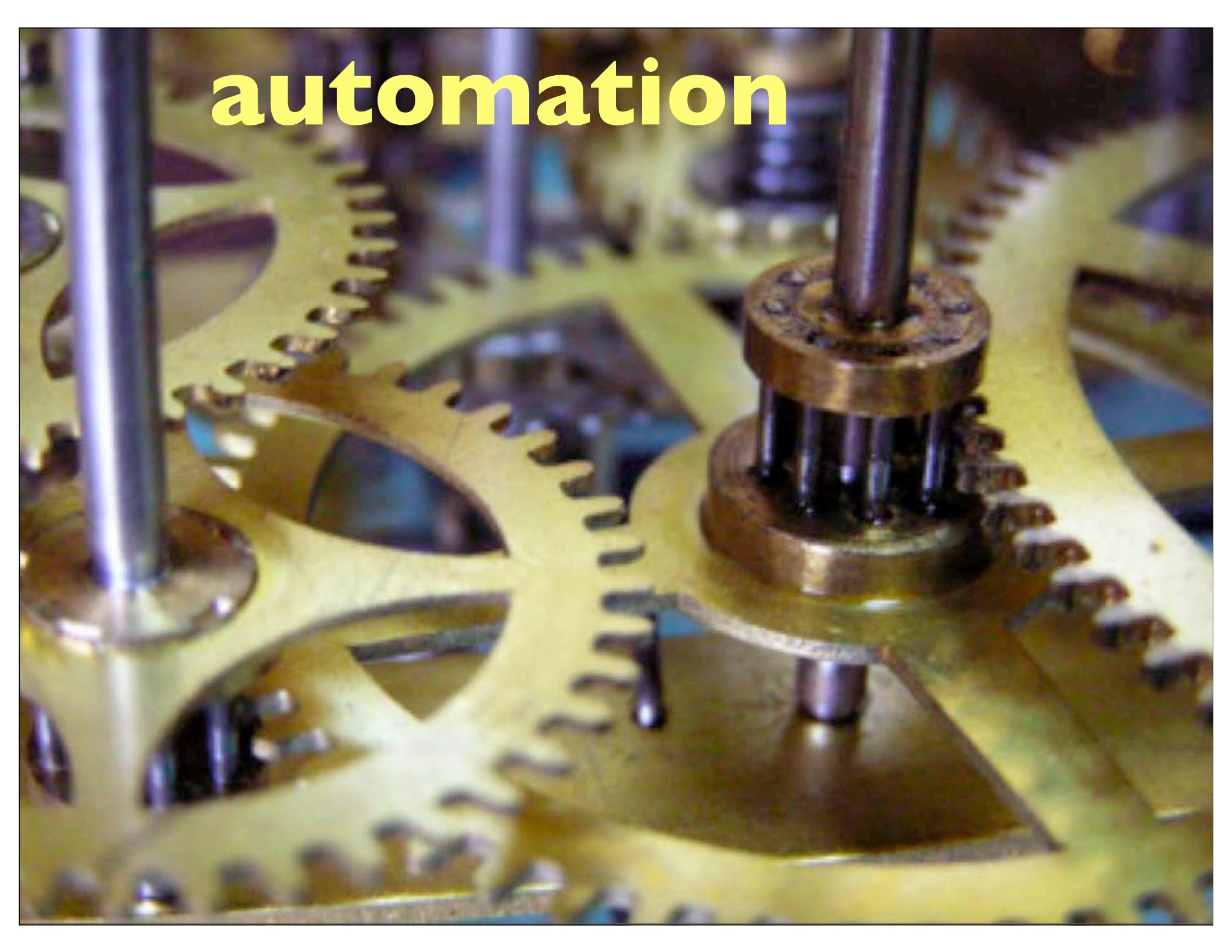
**don't type the same  
commands over and over**



# office 2007/8

excellent  
keyboard  
shortcut  
support



A close-up photograph of a mechanical assembly featuring several interlocking gears and a central shaft. The gears are made of a light-colored metal and are mounted on a dark, possibly black, base. A vertical metal rod or shaft extends from the center of the assembly. The word "automation" is overlaid in a large, bold, yellow sans-serif font in the upper left corner of the image.

automation

# obvious automatables

one-command build

continuous integration

version control (!)

documentation

don't build what you don't  
have to build



**buildix**



open source project from **ThoughtWorks**

infrastructure in a box

live cd or ubuntu installation

# **buildix parts**

**subversion**

**cruisecontrol**

**trac**

**mingle**

# dbDeploy

migrations for the rest of the world

jolt-award winner open source from  
**ThoughtWorks** london

version database schema changes along with  
code

builds delta scripts for you

# using dbDeploy

someone (dba) creates a sql baseline

developers create delta scripts for each change

dbDeploy compares delta with patch table

generates change scripts

invoked from ant's sql task

use migrations to manage  
database schema changes



# yahoo pipes



interactive rss feed aggregator & manipulator

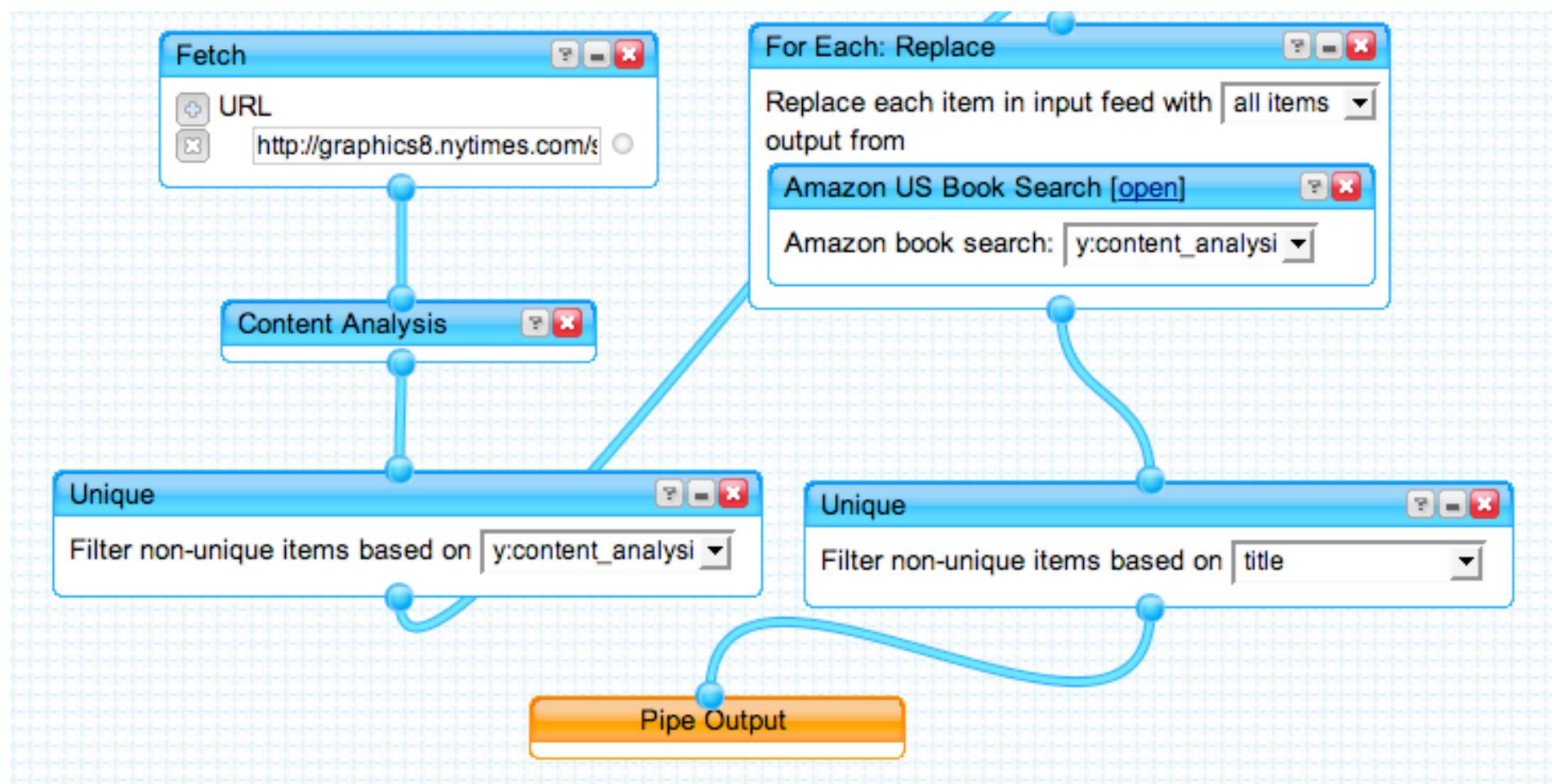
allows you to

combine feeds

manipulate results

aggregate & consolidate feed information

# ny times books



## new york times front page related books

[Run this Pipe](#)    [Clone](#)



Tom Hughes

Amazon search results for keywords generated by content analysis of NY Times front page feed. 3 results per article.

### Pipe Preview

[Subscribe ▾](#)

[How to Interview Like a Top MBA: Job-Winning Strategies From Headhunters, Fortune 100 Recruiters, and Career Counselors](#)  
Array

[How to Become CEO: The Rules for Rising to the Top of Any Organization](#)  
Array

[Competency-Based Resumes: How To Bring Your Resume To The Top Of The Pile](#)  
Array

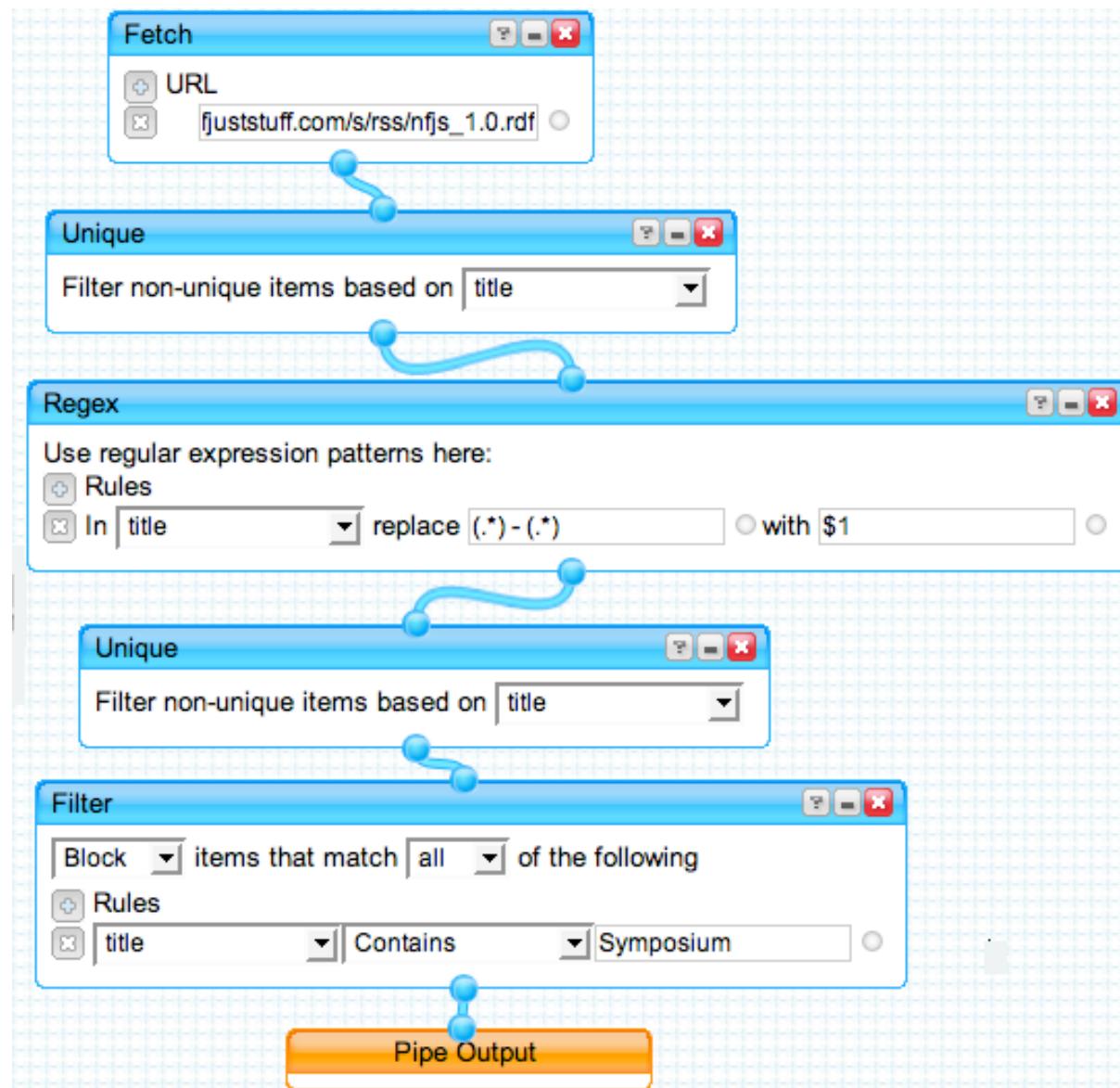
[The Quotable Atheist: Ammunition for Non-Believers, Political Junkies, Gadflies, and Those Generally Hell-Bound](#)  
Array

[Winchester: An American Legend : The Official History of Winchester Firearms and Ammunition from 1849 to the Present](#)  
Array

[The Gun Digest Book Of .22 Rimfire: Rifles, Pistols, Ammunition](#)  
Array

[Fountain Pens of the World](#)  
Array

# recent nfjs bloggers



## Pipe Preview

[Andrew Glover](#)

[Andy Hunt](#)

[Howard Lewis Ship](#)

[Nathaniel Schutta](#)

[Mark Fisher](#)

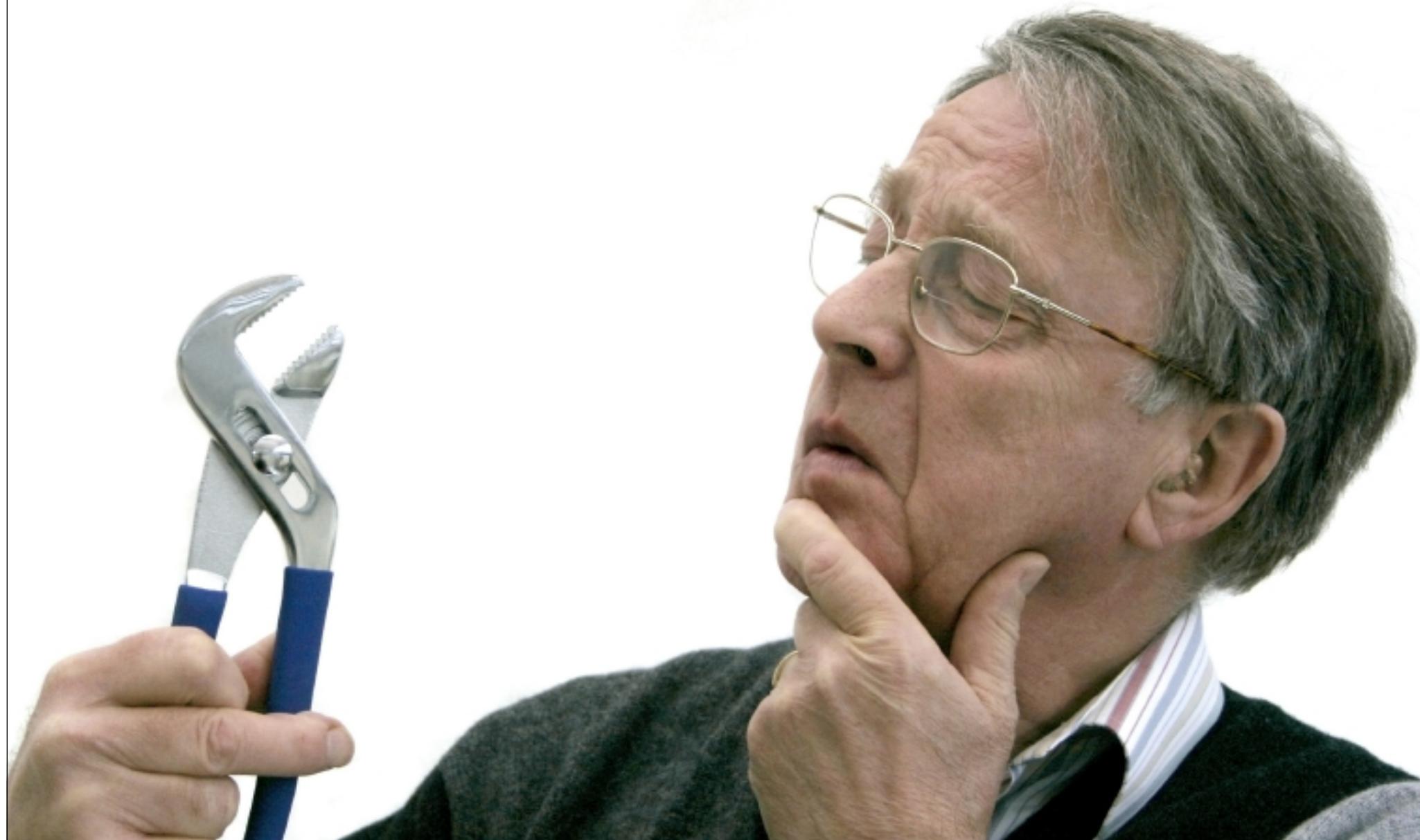
[Ben Hale](#)

[Brian Pontarelli](#)

[Paul Duvall](#)

**Tools:** [!\[\]\(0ae0d7098e1824061ea5df9c7417a653\_img.jpg\) Get as RSS](#) [!\[\]\(510cfb44ca33b1278e8e91a96ea84a71\_img.jpg\) Get as JSON](#)

# subverting other tools



Selenium IDE \*

Base URL

Run  Walk  Step       

**Table** **Source**

Command	Target	Value
type	qty2	3
clickAndWait	submit2	
clickAndWait	returnLink	
type	qty3	2
clickAndWait	submit3	
type	ccNum	345990340934
select	ccType	label=Amex
type	ccExp	3434
clickAndWait	//input[@value='Che...']	

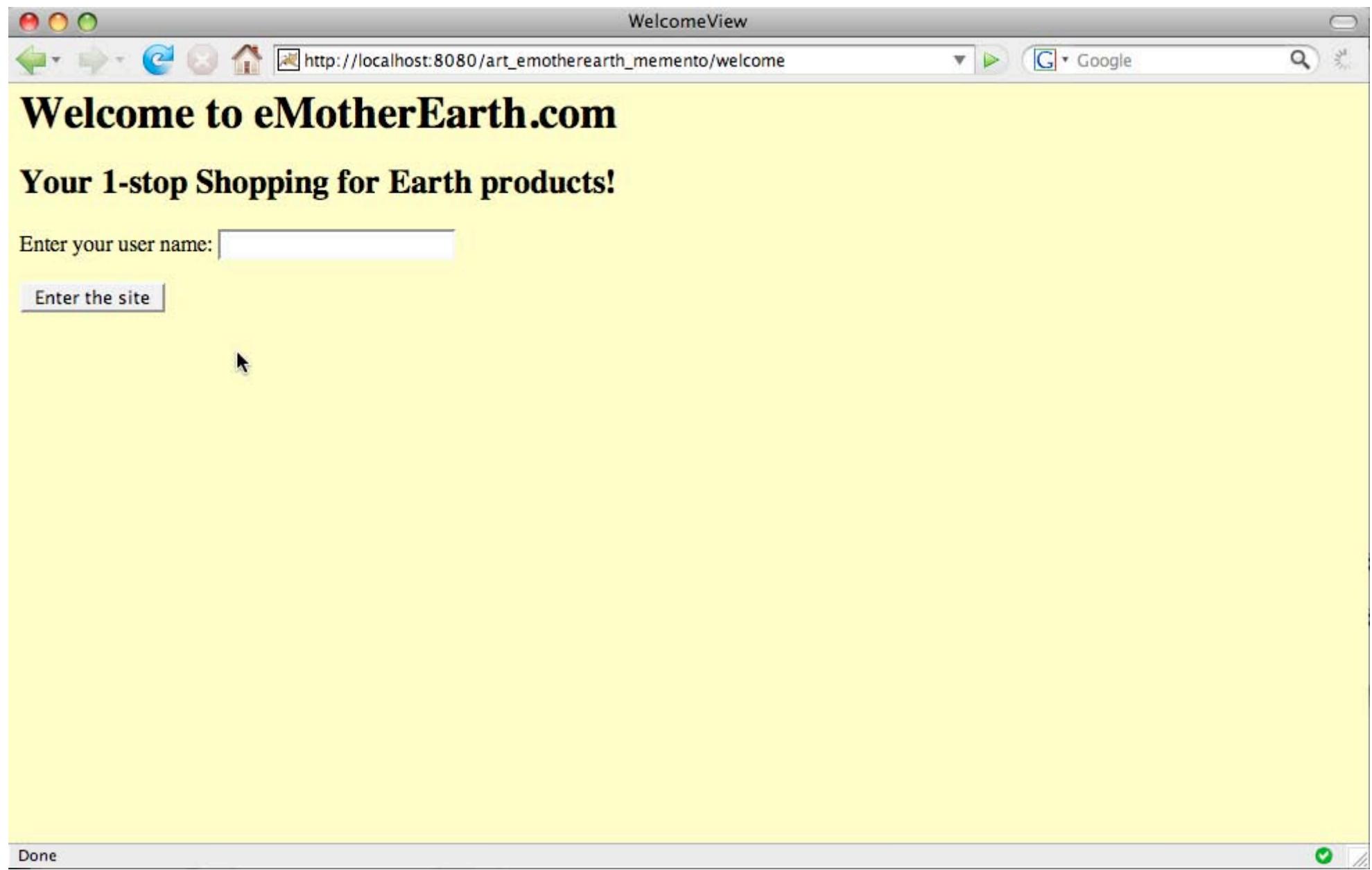
Command

Target  

Value

Log Reference Info ▾ Clear

[info] element has initMouseEvent  
[info] Executing: |type | ccNum | 345990340934 |  
[info] Executing: |select | ccType | label=Amex |  
[info] Executing: |type | ccExp | 3434 |



New Test		
open	/art_emotherearth_memento/welcome	
type	userName	Homer
clickAndWait	submitButton	
type	qty2	3
clickAndWait	submit2	
clickAndWait	returnLink	
type	qty6	4
clickAndWait	submit6	
type	ccNum	234234234234
select	ccType	label=MC
type	ccExp	2323
clickAndWait	//input[@value='Check out']	

```
public class NewTest extends SeleneseTestCase {
    public void testNew() throws Exception {
        selenium.open("/art_emotherearth_memento/welcome");
        selenium.type("userName", "Homer");
        selenium.click("submitButton");
        selenium.waitForPageToLoad("30000");
        selenium.type("qty2", "3");
        selenium.click("submit2");
        selenium.waitForPageToLoad("30000");
        selenium.click("returnLink");
        selenium.waitForPageToLoad("30000");
        selenium.type("qty6", "4");
        selenium.click("submit6");
        selenium.waitForPageToLoad("30000");
        selenium.type("ccNum", "234234234234");
        selenium.select("ccType", "label=MC");
        selenium.type("ccExp", "2323");
        selenium.click("//input[@value='Check out']");
        selenium.waitForPageToLoad("30000");
    }
}
```

```
class NewTest < Test::Unit::TestCase
  def setup
    @verification_errors = []
    if $selenium
      @selenium = $selenium
    else
      @selenium = Selenium::SeleneseInterpreter.new(
        "localhost", 4444, "*firefox", "http://localhost:4444", 10000);
      @selenium.start
    end
    @selenium.set_context("test_new", "info")
  end

  def teardown
    @selenium.stop unless $selenium
    assert_equal [], @verification_errors
  end

  def test_new
    @selenium.open "/art_emotherearth_memento/welcome"
    @selenium.type "userName", "Homer"
    @selenium.click "submitButton"
    @selenium.wait_for_page_to_load "30000"
    @selenium.type "qty2", "3"
    @selenium.click "submit2"
    @selenium.wait_for_page_to_load "30000"
    @selenium.click "returnLink"
    @selenium.wait_for_page_to_load "30000"
    @selenium.type "qty6", "4"
    @selenium.click "submit6"
    @selenium.wait_for_page_to_load "30000"
    @selenium.type "ccNum", "234234234234"
    @selenium.select "ccType", "label=MC"
    @selenium.type "ccExp", "2323"
    @selenium.click "//input[@value='Check out']"
    @selenium.wait_for_page_to_load "30000"
  end
end
```

Default Default Default

```
13:42:44.000 INFO - Command request: waitForPageToLoad[30000, ] on session 554644
13:42:44.089 INFO - Got result: OK on session 554644
13:42:44.102 INFO - Command request: isElementPresent[restoreButton, ] on session 554644
13:42:44.115 INFO - Got result: OK on session 554644
13:42:44.117 INFO - Command request: click[restoreButton, ] on session 554644
13:42:44.147 INFO - Got result: OK on session 554644
13:42:44.149 INFO - Command request: isElementPresent[restoreButton, ] on session 554644
13:42:44.178 INFO - Got result: OK on session 554644
13:42:44.181 INFO - Command request: click[restoreButton, ] on session 554644
13:42:44.205 INFO - Got result: OK on session 554644
13:42:44.206 INFO - Command request: isElementPresent[restoreButton, ] on session 554644
13:42:44.248 INFO - Got result: OK on session 554644
13:42:44.252 INFO - Command request: click[restoreButton, ] on session 554644
13:42:44.329 INFO - Got result: OK on session 554644
13:42:44.330 INFO - Command request: isElementPresent[restoreButton, ] on session 554644
13:42:44.394 INFO - Got result: OK on session 554644
13:42:44.395 INFO - Command request: click[submit, ] on session 554644
13:42:44.435 INFO - Got result: OK on session 554644
13:42:44.438 INFO - Command request: isElementPresent[submit, ] on session 554644
13:42:44.438 INFO - Killing browser
13:42:44.616 INFO - Got result: OK on session 554644
```

OrderDbImproved.java RandomQuantitiesTest.java ShowCart.java OrderDbBest.java ShoppingCart.java

```
public void test_Random_number_of_quantities() throws Exception {
    s.open("/art_emotherearth_memento/RawData.jsp");
    String userName = s.getValue("user_name");
    String ccNum = s.getValue("cc_num");
    String ccType = s.getValue("cc_type");
    String ccExpDate = s.getValue("cc_exp_date");
    s.open("/art_emotherearth_memento/welcome");
    assertEquals("WelcomeView", s.getTitle());
    s.type("userName", userName);
    s.click("submitButton");
    s.waitForPageToLoad("30000");
    assertEquals("CatalogView", s.getTitle());
    int randomNumberOfInvocations = random.nextInt(5) + 1;
    for (int invocations = 0; invocations < randomNumberOfInvocations; invocations++) {
        int randomQuantity = random.nextInt(3) + 1;
        int randomProductId = random.nextInt(5) + 1;
        s.type("qty" + randomProductId, String.valueOf(randomQuantity));
        s.click("submit" + randomProductId);
    }
}
```

Run Test Suite

- DecisionDemoTest (com.nealford.art.memento.emotherearth)
  - test\_Simple\_transaction\_runs\_from\_first\_to\_last\_page
  - test\_Undo\_operation\_restores\_button\_state\_and\_shopping

Done: 2 of 2(18.91 s)

Output Statistics

Process finished with exit code 0

Compilation completed successfully 43:67 Insert Default 149M of 190M

# automated interaction

record your interaction the 1st time you walk through the page

literally cuts hours off debugging time

selenium defines an interaction api for web applications

have your q/a department record bug discoveries

don't spend time doing by  
hand what you can automate



# build your own tools



# build your own tools

you almost never do anything just once

work like a craftsman, not a laborer

build shims & jigs

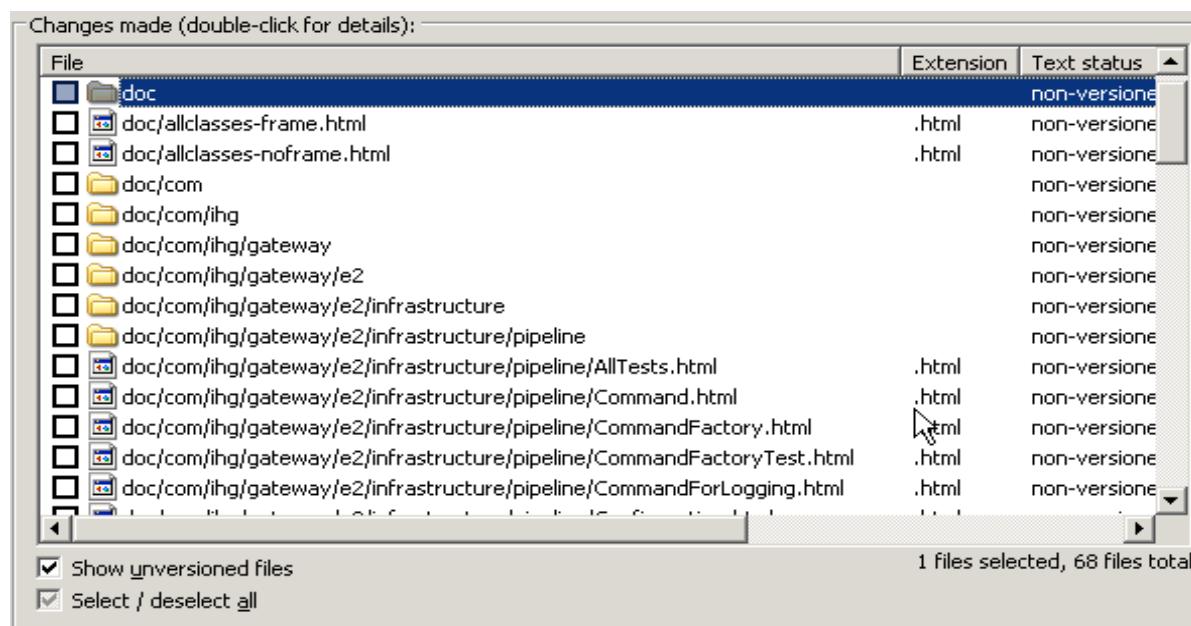
building a tool takes a little longer than brute force...

...but you build assets

# bash-fu

adding new files to subversion repository

tortoise (on windows), but with limits



# svnAddNew

```
svn st | grep '^?*' | tr '?*' ' ' |  
sed 's/ [ ]*//' | sed 's/[ ]/\ /g' | xargs svn add
```

**svn st**

get svn status (new files start with “?”)

**grep '^?\***

find all new files

**tr '?\*' ' '**

translate the “?” into a space

**sed 's/[ ]\*//'**

substitute spaces to nothing

**sed 's/[ ]/\ /g'**

escape embedded spaces

**xargs svn add**

pipe the improved arguments into svn

# more bash-fu

the problem: 2 gb of log files / week

need to know the count of each exception type

by hand?!?

automate with a bash script

get all exception types from log file

```
#!/bin/bash
for X in $(egrep -o "[A-Z]\w*Exception" genesis_week.txt | sort | uniq) ;
do
    echo -n -e "$X\t"
    grep -c "$X" genesis_week.txt
done
```

get counts of each  
exception

sort them

get unique  
list

# windows scripting host

```
<Job id="dailyLogs">
<script language="JScript">
    //--- build list of files
    S_DOCS = "c:\\jNf\\";
    aFiles = new Array();
    aFiles[aFiles.length] = S_DOCS + "im\\2005 Weekly Workout Planner.xls";
    aFiles[aFiles.length] = S_DOCS + "im\\2004 Workout Log.xls";
    aFiles[aFiles.length] = S_DOCS + "trackers\\Weight Tracker.xls";

    //--- get reference to Excel objects
    var oExcelApp, oExcelWorkBooks;
    oExcelApp = new ActiveXObject("Excel.Application");
    oExcelWorkbooks = oExcelApp.WorkBooks;

    //--- automate Excel
    oExcelApp.Visible = true;
    //--- open the files
    for (iIndex = 0; iIndex < aFiles.length; iIndex++ ) {
        oExcelWorkbooks.open(aFiles[iIndex], true);
    }
    oExcelApp.Windows.Arrange(7);

</Script>
</Job>
```

# windows power shell

powerful replacement for windows scripting host

uses objects the way \*-nix uses plain text

pipe properties from 1 command to another

```
dir | where-object { $_.LastWriteTime -like "6/1/2006 *" } |  
move-item -destination c:/DestFolder
```

# comparisons

bash    `ps -e | grep " p" | awk '{ print $1 }' | xargs kill`

wps    `get-process p* | stop-process`

bash    `ps -el | awk '{ if ( $6 > (1024*10)) { print $3 } }' | grep -v PID | xargs kill`

wps    `get-process | where { $_.VS -gt 10M } | stop-process`

# implemented with .net

use .net types within power shell scripts

```
get-member -input "String" -membertype method
```

# cmdlets

Get-Command	Retrieves a list of all available cmdlets.
Get-Help	Displays help information about cmdlets and concepts.
Get-WMIObject	Retrieves management information by using WMI.
Get-EventLog	Retrieves Windows event logs.
Get-Process	Retrieves a single or list of active processes.
Get-Service	Retrieves a Windows service.
Get-Content	Reads in text files, treating each line as a child object.
Add-Content	Appends content to a text file.
Copy-Item	Copies an item
Get-Acl	Retrieves access control lists (ACLs).

# just one thing...

windows power shell implemented in .net

take advantage of .net types and libraries

.net coupled to every nook & cranny in windows

“headless” longhorn arriving rsn

won’t have windows power shell!

# automating com

```
def open_daily_logs
    excel = WIN32OLE.new("excel.application")

    workbooks = excel.WorkBooks
    excel.Visible = true
    doc_list.each do |f|
        begin
            workbooks.Open(@@Home_Dir + f, true)
        rescue
            puts "Cannot open workbook:", @@Home_Dir + f
        end
    end
    excel.Windows.Arrange(7)
end
```

# scripting rationale

examples in lots of different languages/tools

which one do I use for this problem?

use a *real* language for scripting

# sql splitter

the problem: split a 38,000 line sql file into  
1000 line chunks

each chunk must be syntactically correct

“we can do it by hand in 10 minutes...”

automate instead

after 50 minutes:

```
SQL_FILE = "./GeneratedTestData.sql"
OUTPUT_PATH = "./chunks of sql/"

line_num = 1
file_num = 0
Dir.mkdir(OUTPUT_PATH) unless File.exists? OUTPUT_PATH
file = File.new(OUTPUT_PATH + "chunk " + file_num.to_s + ".sql",
               File::CREAT|File::TRUNC|File::RDWR, 0644)

done, seen_1k_lines = false
IO.readlines(SQL_FILE).each do |line|
  file.puts(line)
  seen_1k_lines = (line_num % 1000 == 0) unless seen_1k_lines
  line_num += 1
  done = (line.downcase =~ /^W*goW*/ or
          line.downcase =~ /^W*endW*/) != nil
  if done and seen_1k_lines
    file_num += 1
    file = File.new(OUTPUT_PATH + "chunk " + file_num.to_s + ".sql",
                   File::CREAT|File::TRUNC|File::RDWR, 0644)
    done, seen_1k_lines = false
  end
end
```

# time spent automating

it took us 5 times longer to automate it

we've had to do it numerous times since

it “accidentally” became an important part of  
our project

using a real language allowed us to refactor it...

...so that we could write unit tests

```
def test_mocked_out_dir
  ss = SqlSplitter.new("dummy_path", "dummy_file")
  Dir.expects(:mkdir).with("dummy_path")
  ss.make_a_place_for_output_files
end

def test_that_output_directory_is_created_correctly
  ss = SqlSplitter.new(OUTPUT_PATH, nil)
  ss.make_a_place_for_output_files
  assert File.exists? OUTPUT_PATH
end

def test_that_lines_o_sql_has_lines_o_sql
  lines = %w{Lorem ipsum dolor sit amet consectetur}
  ss = SqlSplitter.new(nil, nil)
  ss.sql_lines = lines
  assert ss.lines_o_sql.size > 0
  assert_same ss.lines_o_sql, lines
end

def test_generate_sql_chunks
  ss = SqlSplitter.new(OUTPUT_PATH, nil)
  ss.sql_lines = lots_o_fake_data
  ss.generate_sql_chunks
  assert File.exists? OUTPUT_PATH
  assert Dir.entries(OUTPUT_PATH).size > 0
  Dir.entries(OUTPUT_PATH).each do |f|
    assert f.size > 0
  end
end
```

# using real languages

allow throw-aways to grow into assets

allows unit testing, refactoring, ide support

if you start by treating it as a 1st class problem,  
you'll build better solutions

# time savings

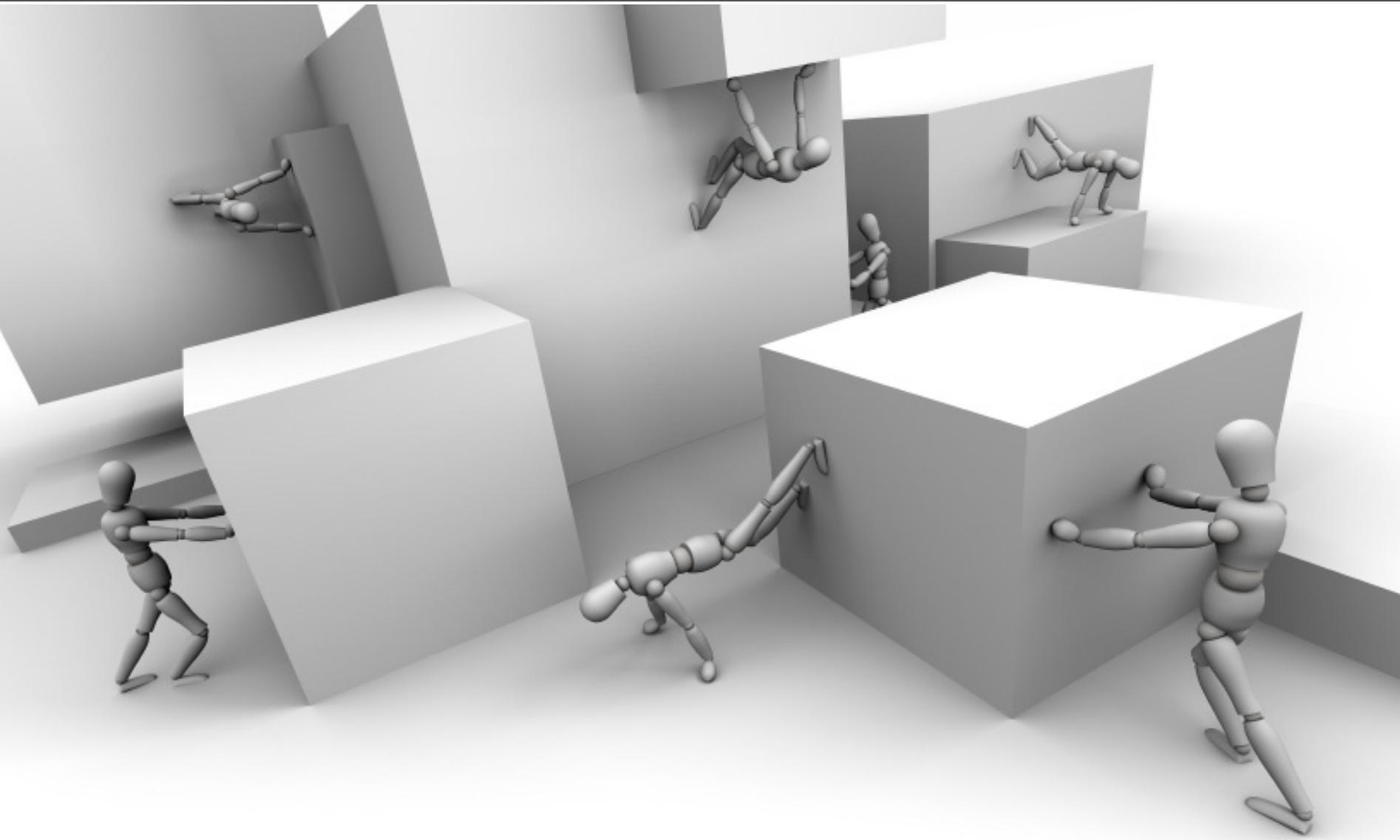
solving problems by hand makes you dumber

steals concentration

squanders focus

automating makes you smarter

figure out clever ways to solve problems



# justifying automation

# timebox

set a reasonable time to see if it's possible

evaluate at the end of the box

decide if you want to go forward

or create another time box

or abandon the effort

# analyze the r.o.i.

how long does it take now X # of times we must do it?

what are the consequences of doing it wrong 1 time?

automation is about

time savings

risk mitigation

**don't shave yaks**



A photograph of a person meditating in a cross-legged, lotus-like position on a rock or log by a lake. The person is silhouetted against the warm, orange and yellow hues of a setting sun reflected in the water. The background shows a dark, silhouetted shoreline with trees.

focus

# simple stuff

get a comfortable chair!

dual monitors...

...sitting immediately in front of you

administrator privilege for the o/s

good keyboard

# insidious distractions

modern office environments are terrible for  
knowledge workers

too much out of context noise

how many people here work in...

# war rooms



# locus of attention

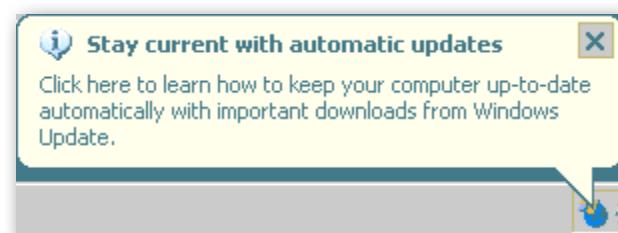
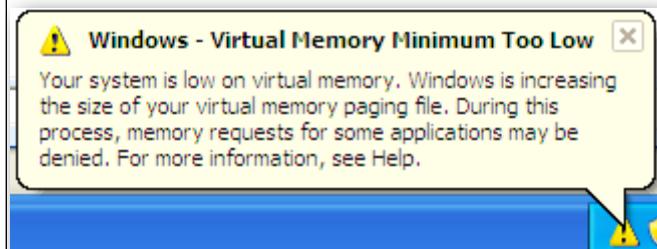
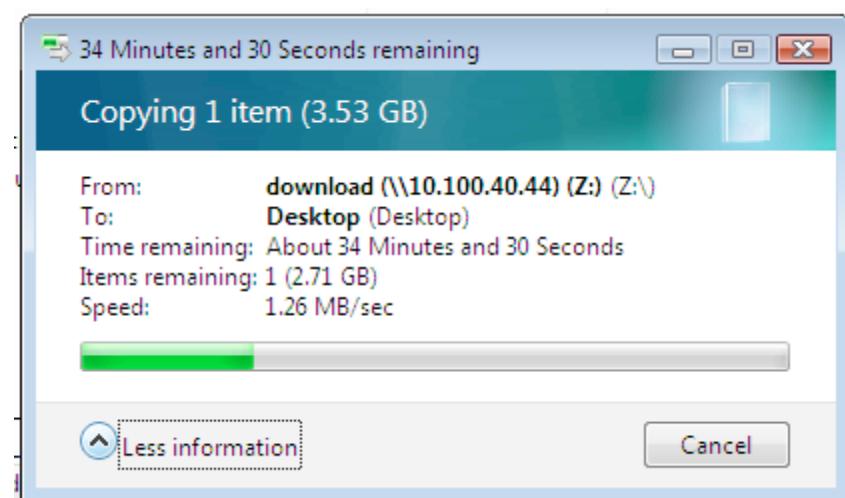
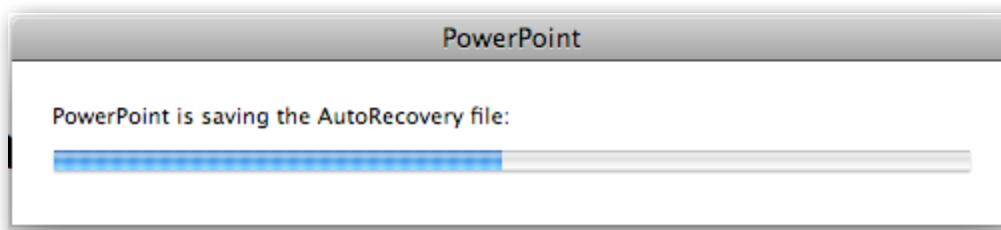
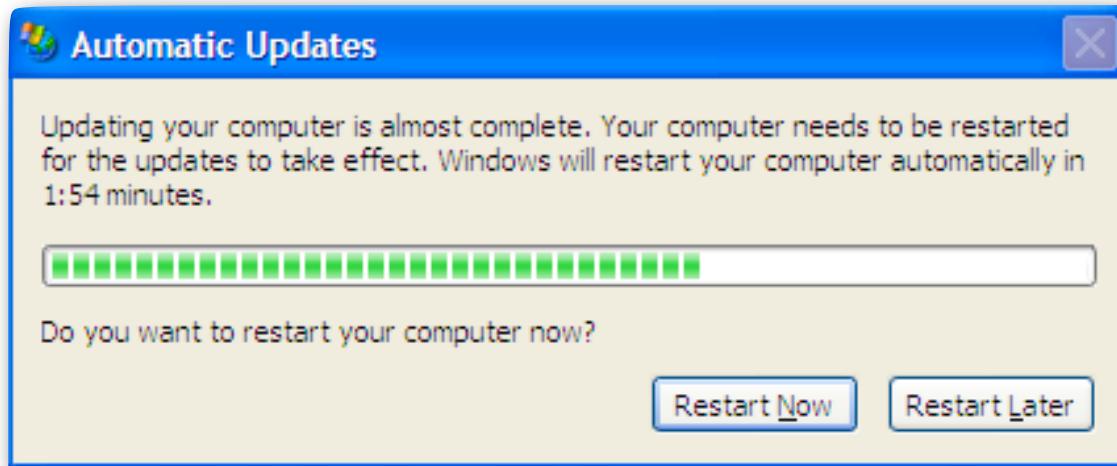
in *the humane interface*, jef raskin describes *locus of attention*

anything that happens outside your locus of attention breaks *flow*

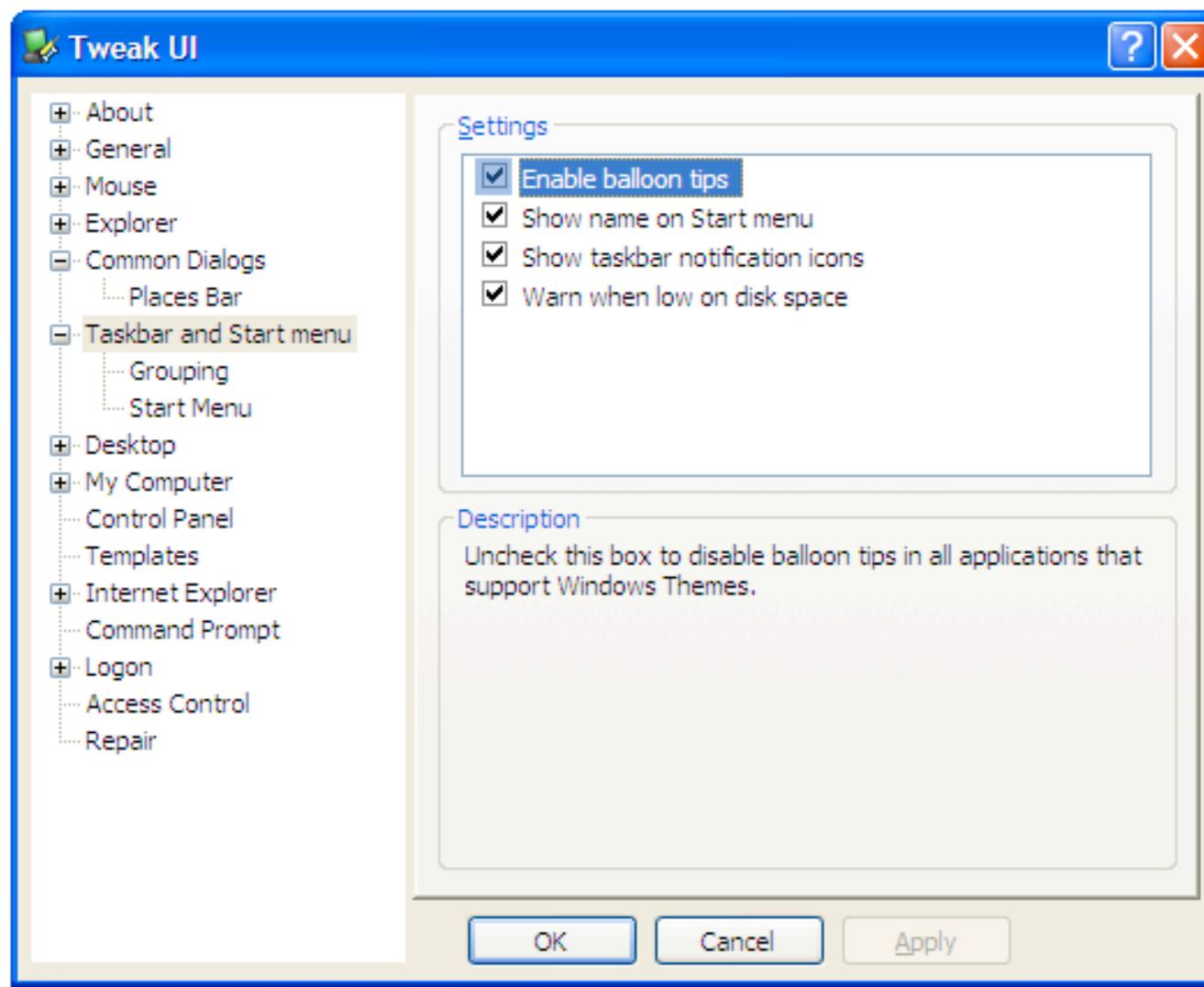
in *flow*, michael csikszentmihalyi describes flow state

total concentration

time disappears



# killing balloon tips



# screen dimmers

automatically makes your background dark  
after a set time

jedi concentrate



doodim



the higher the level of  
concentration, the denser  
the ideas



# the easy stuff

turn off notifications

don't keep email open

turn off instant messaging

put on headphones

create office “quiet time”

# focus techniques



# search > navigation

all developer hierarchies are too deep

file system

package/namespace

documentation

what worked well with 20 mb hard drives fails  
with 200 gb

# desktop search

built into modern operating systems

retro-fittable in older ones

google desktop search

larry's “any text file” indexer

**replace file hierarchy  
navigation with search**



# **find + grep**

**from \*-nix & cygwin**

**find** searches for files within a hierarchy

**grep** searches for strings from input or files

from the current  
directory

pipe through

xargs

matching this text

```
find . -name "*.java" | xargs grep -n -H "new .*Db.*"
```

matching files  
named \*.java

grep with filenames  
& line numbers

chain together conditions



```
find -name "*.java" -not -regex ".*Db\\.java"  
      -exec grep -H -n "new .*Db" {} \\;
```

execute the following on  
all matched files

placeholder for file name

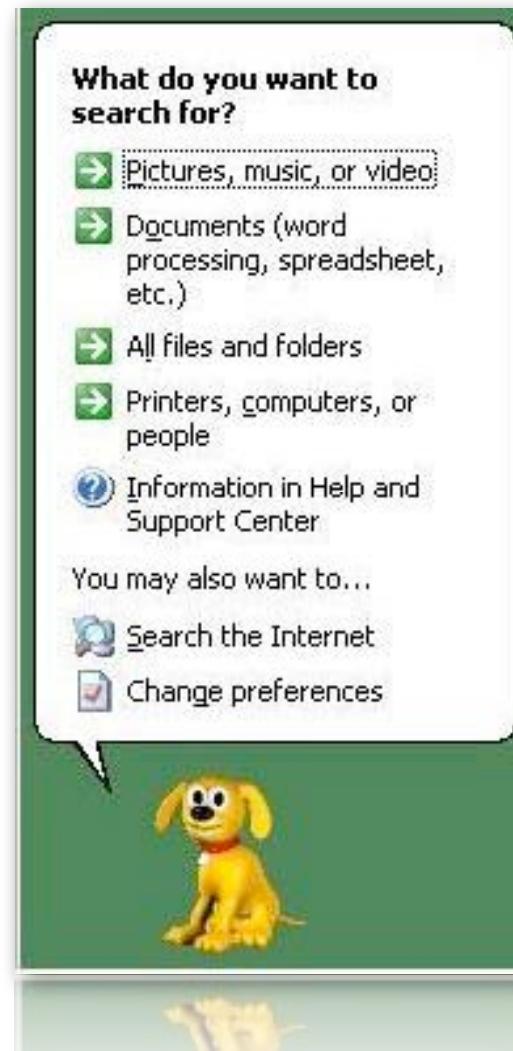
terminate  
the exec  
command

```
find -name *.java -exec egrep -n -H  
"(public|private).*\b(ShoppingCart\b.*" {} \;
```

use “hard target” searches  
to query your code base



# what find + grep can't provide



# rooted views

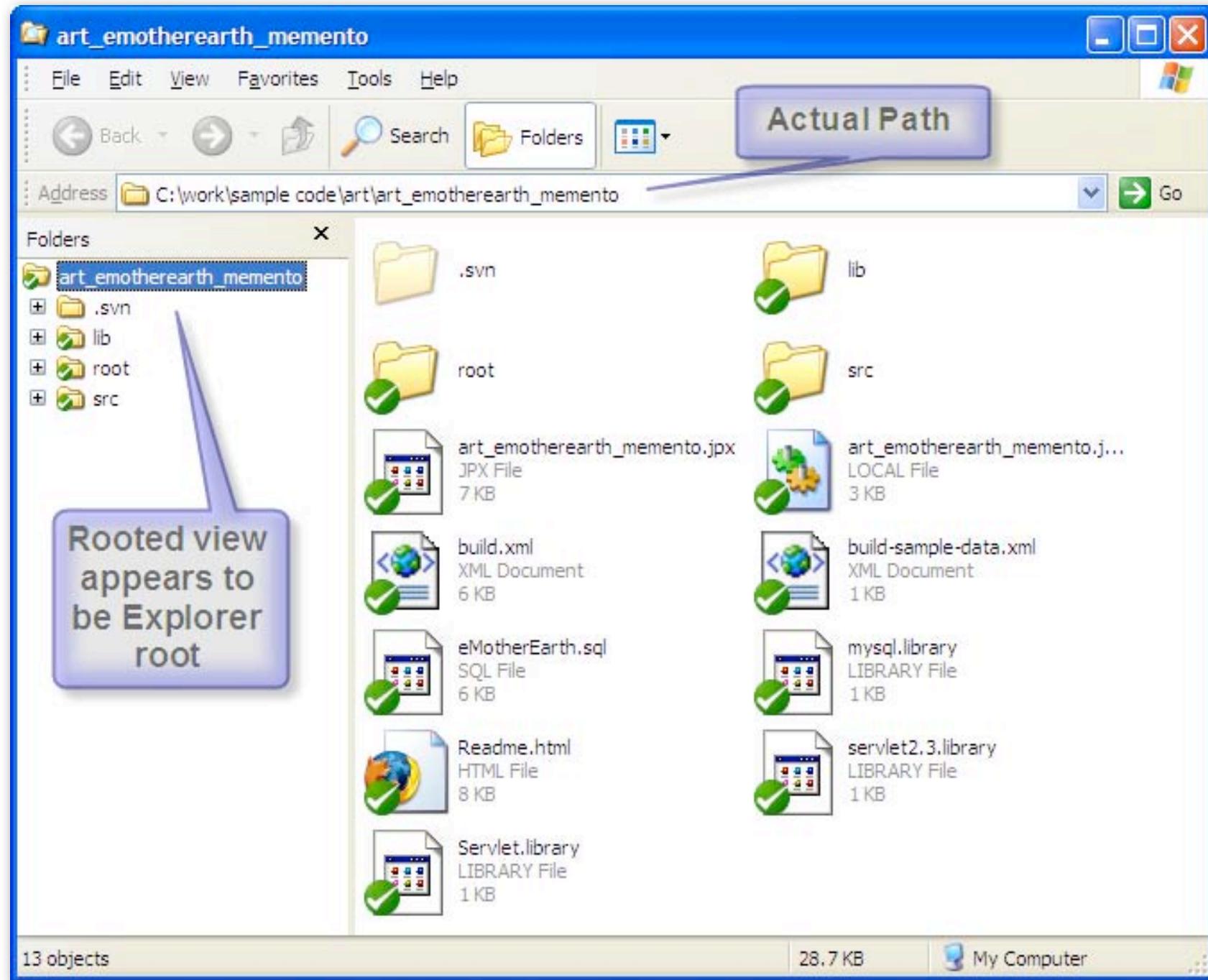
specialized explorer view

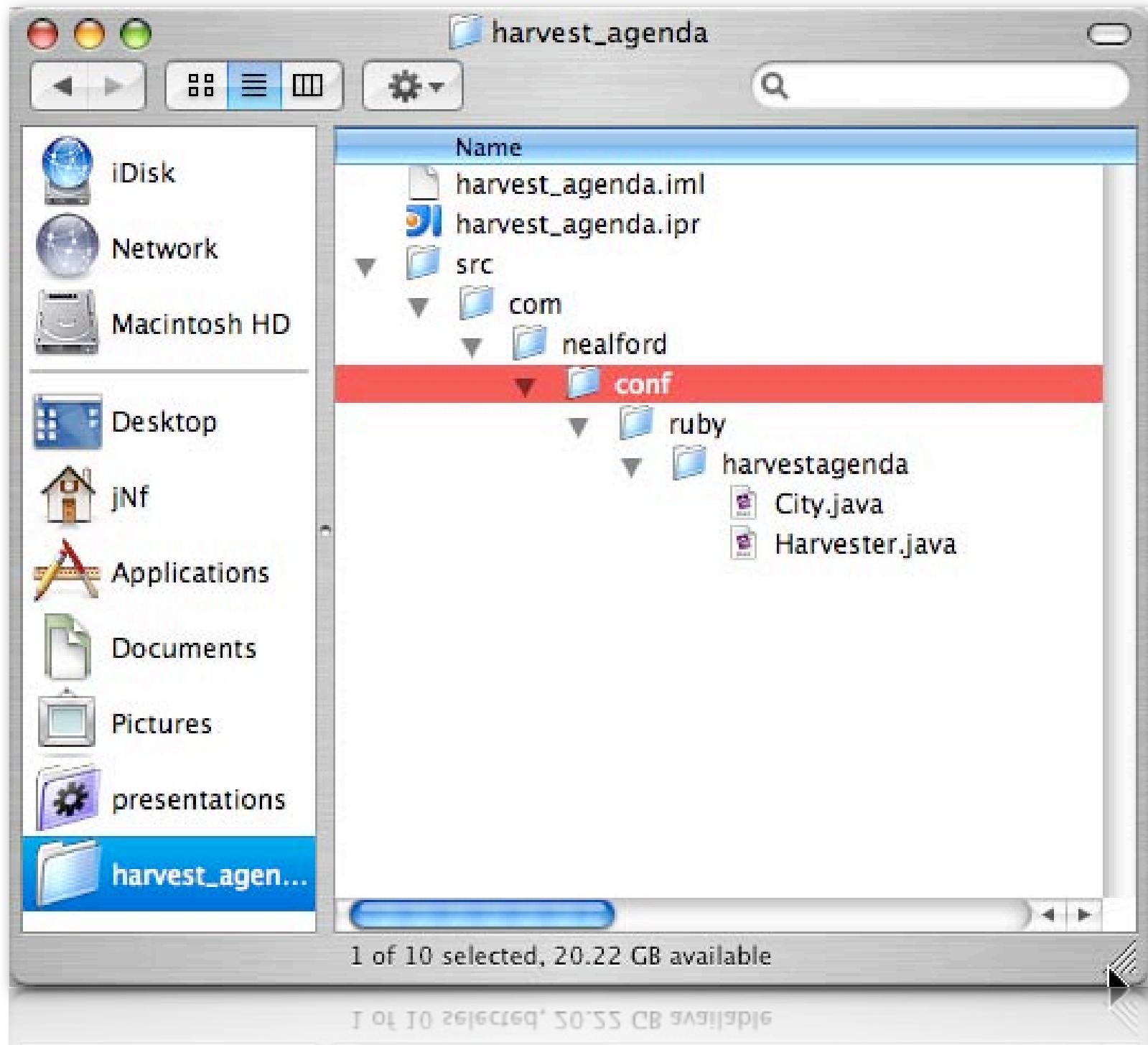
especially good for directory-based version control

rooted view == project explorer

create a shortcut:

```
C:\WINDOWS\explorer.exe /e,/root,c:\work\project
```





# sticky attributes

for command windows

windows keeps customized settings keyed to  
the title of the window

when you close the window, you can save the  
settings

# **useful sticky attributes**

<b>almost infinite scrolling</b>	<b>default is 300 make it 9999</b>
<b>screen width</b>	<b>as wide as you can make it without horizontal scrolling</b>
<b>unique position</b>	<b>helps identify its purpose</b>
<b>unique colors</b>	<b>set foreground and background colors based on use</b>
<b>quick edit mode</b>	<b>turn it off</b>

# project-based shortcuts

use links to create “smart directories” for specific projects

project folders on quick launch bar



stacks



# use virtual desktops

virtual desktops allow you multiple isolated environments

bind applications to desktops

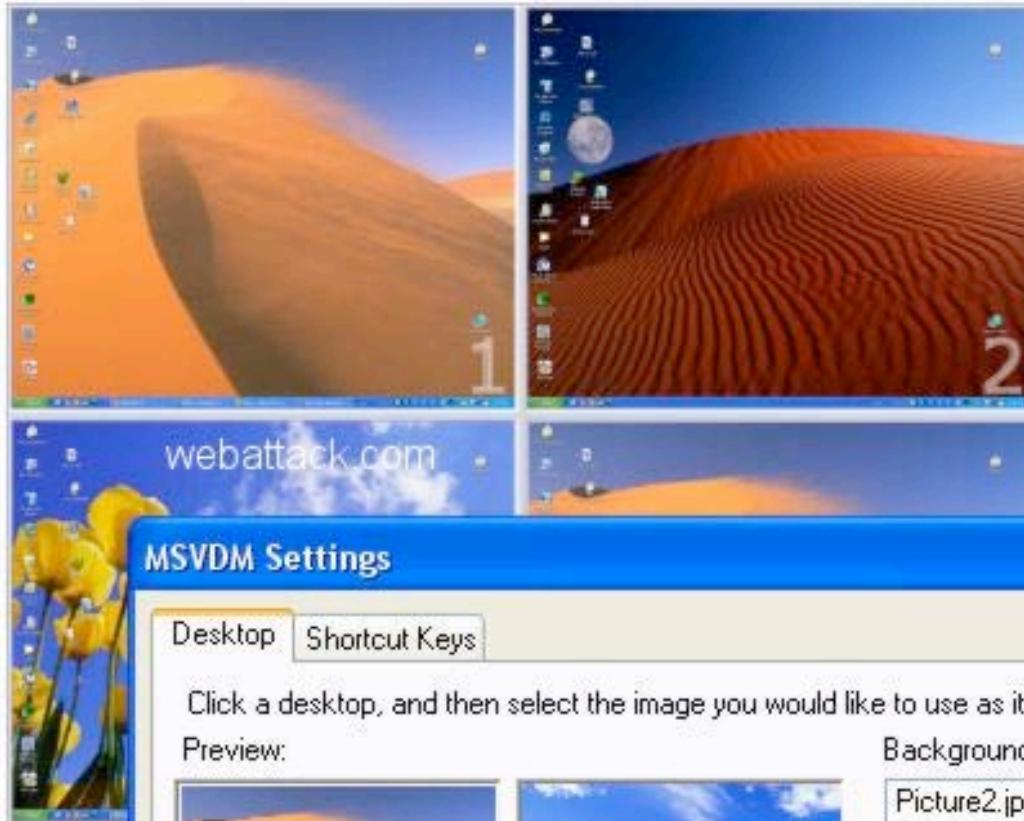
virtual desktop manager power toy



virtue desktops (in tiger)



spaces (in leopard)



### MSVDM Settings

Desktop   Shortcut Keys

Click a desktop, and then select the image you would like to use as its background.

Preview:



Background:

- Picture2.jpg
- Power.jpg
- Purple flower.jpg
- Radiance.jpg
- Red moon desert.jpg
- Ripple.jpg
- step1\_focus.gif
- step2.gif
- Stonehenge.jpg
- Tulips.jpg
- Vortec space.jpg
- Wind.jpg

Position:

Stretch

Browse...

OK

Cancel

Apply

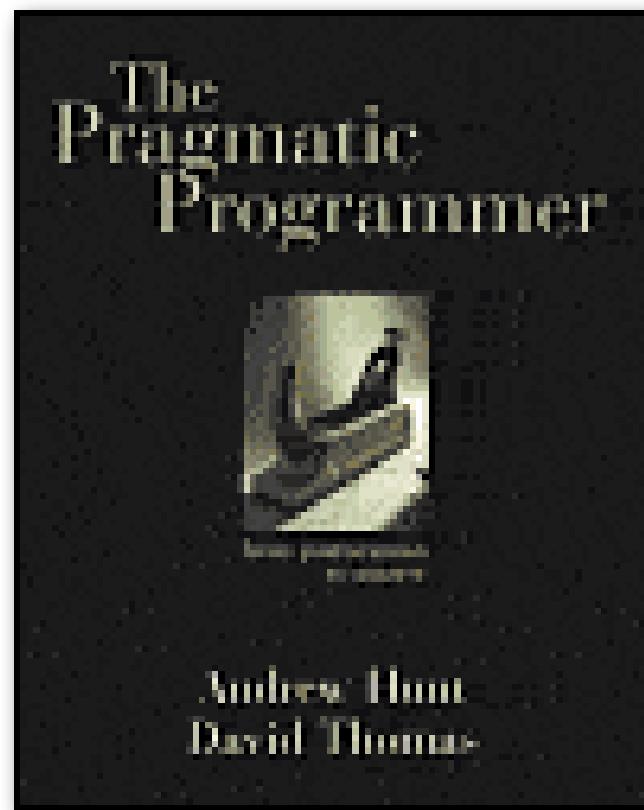
MSVDM

1 2 3 4 ← 3:30 PM

# canonicity



# DRY principle



DRY principle  
proper noun

repetition is the single  
most insidious diminishing  
force in software  
development



i  
n  
d  
i  
r  
e  
c  
t  
i  
o  
n



# indirection

the ability to point to something else

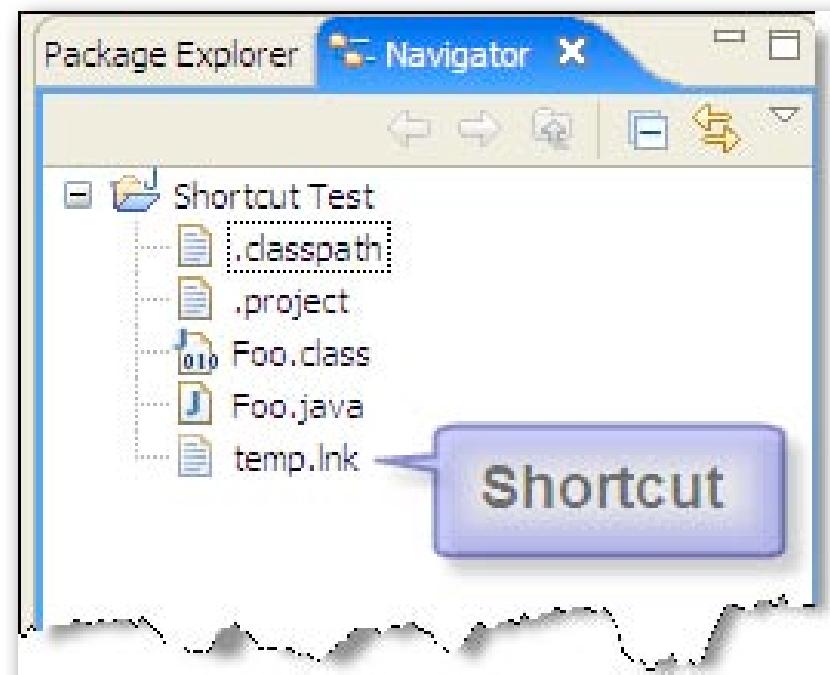
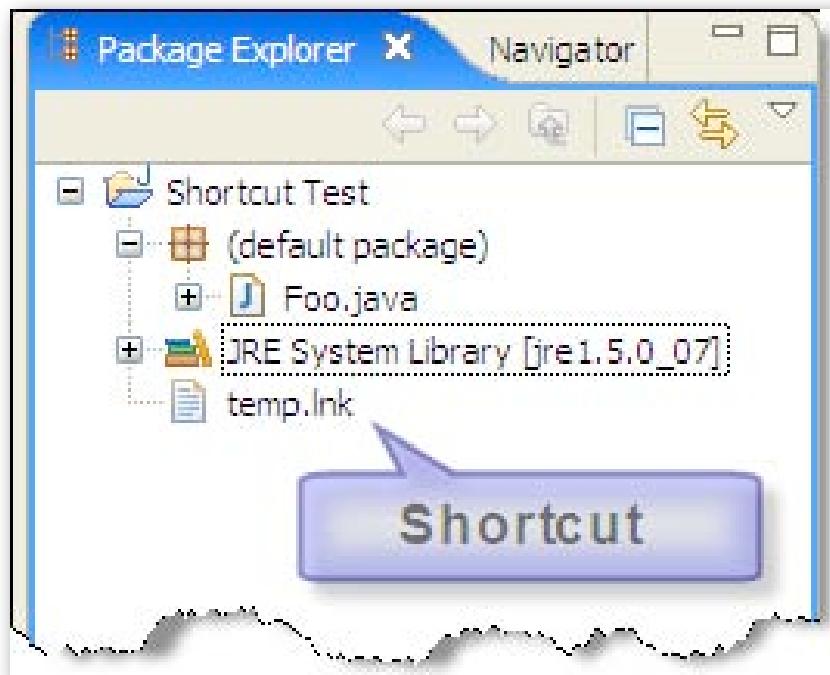
links in \*-nix

unfortunately, not available in windows < vista

shortcuts are just a veneer

they cause “metaphor shear”

# metaphor shear



# junction links

true links for windows

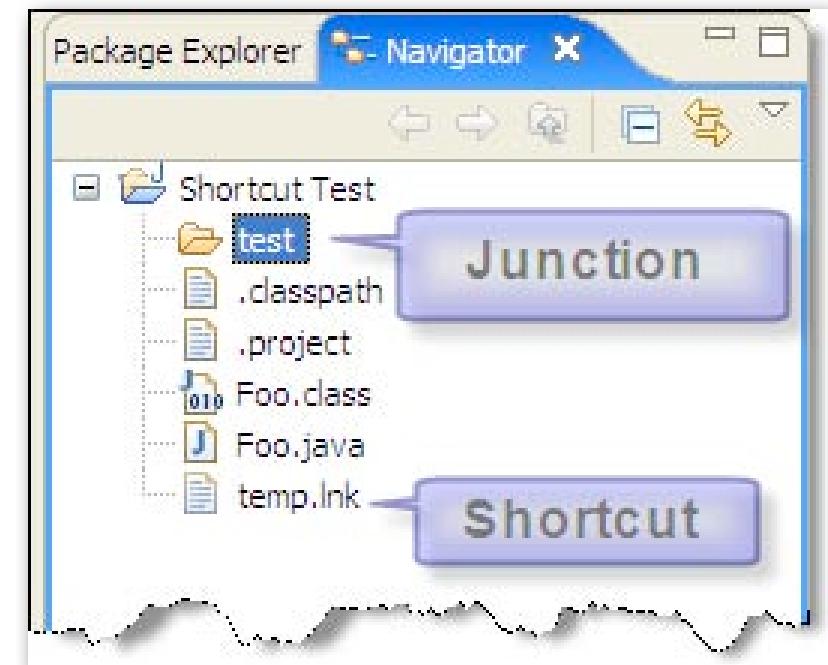
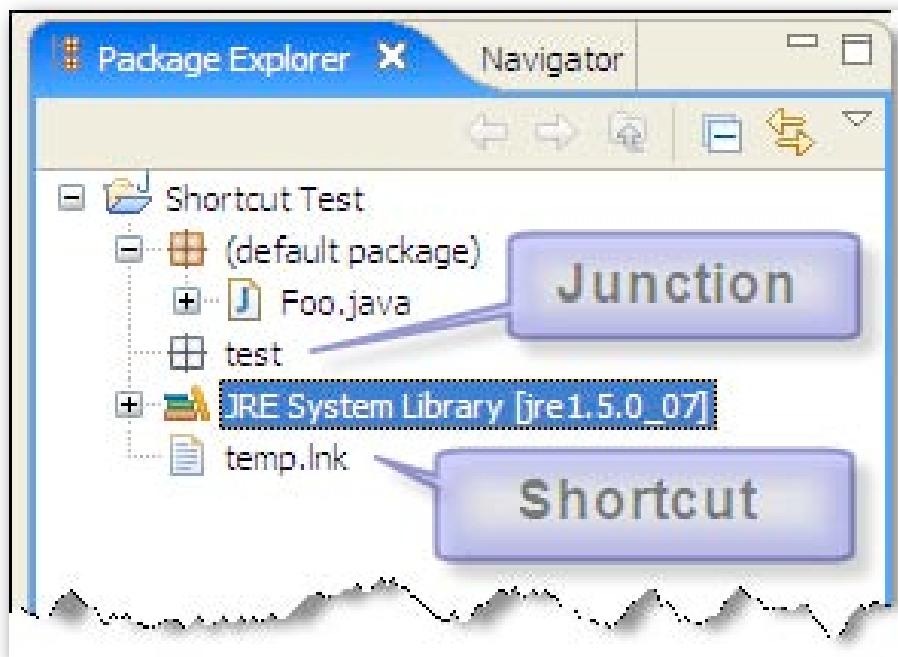
```
junction <target> <source>
```

```
junction myproj \My Documents\Development\Sisyphus\NextGen
```

must be NTFS file system

only links to directories

# junction links



# indirection in action

all documents in a subversion repository

record macros for jedit

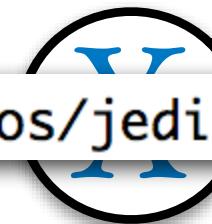
jedit macros must live in `~/.jedit/macros`

home directory isn't in subversion

indirection to the rescue



/Documents/dev/macros/jedit <= <home>/.jedit/macros



/Documents/dev/macros/jedit <= ~/.jedit/macros



\Documents and Settings\inford\.jedit\ macros

~/.jedit\ macros

/Documents/dev/macros/jedit



Subversion  
Repository



\Documents and Settings\ngford\jedit\ macros

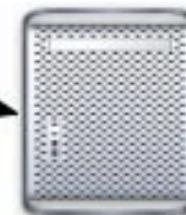
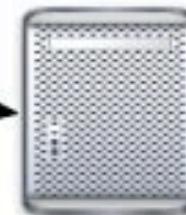
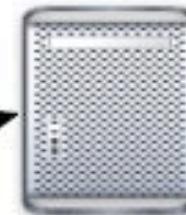
## Subversion Repositories

/Documents/dev/macros/jedit/

Project 1/

Project 2/

Project 3/



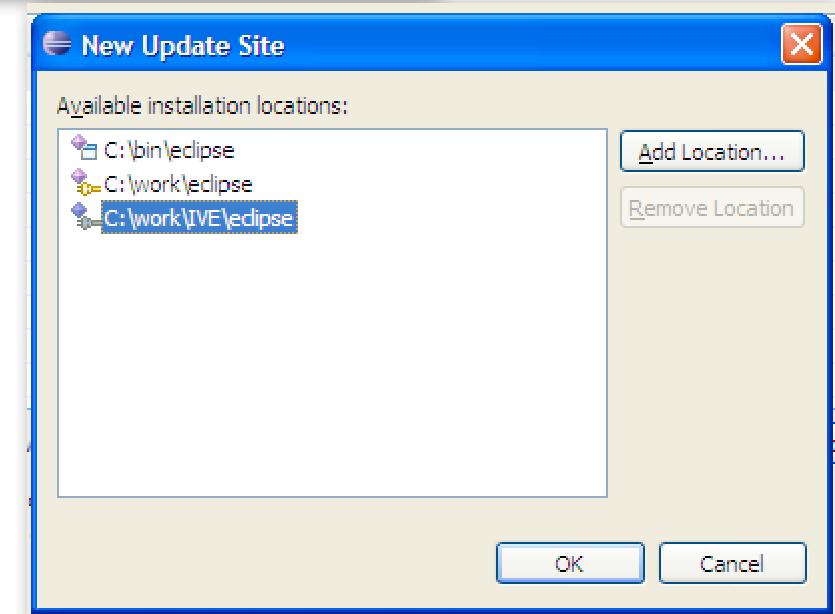
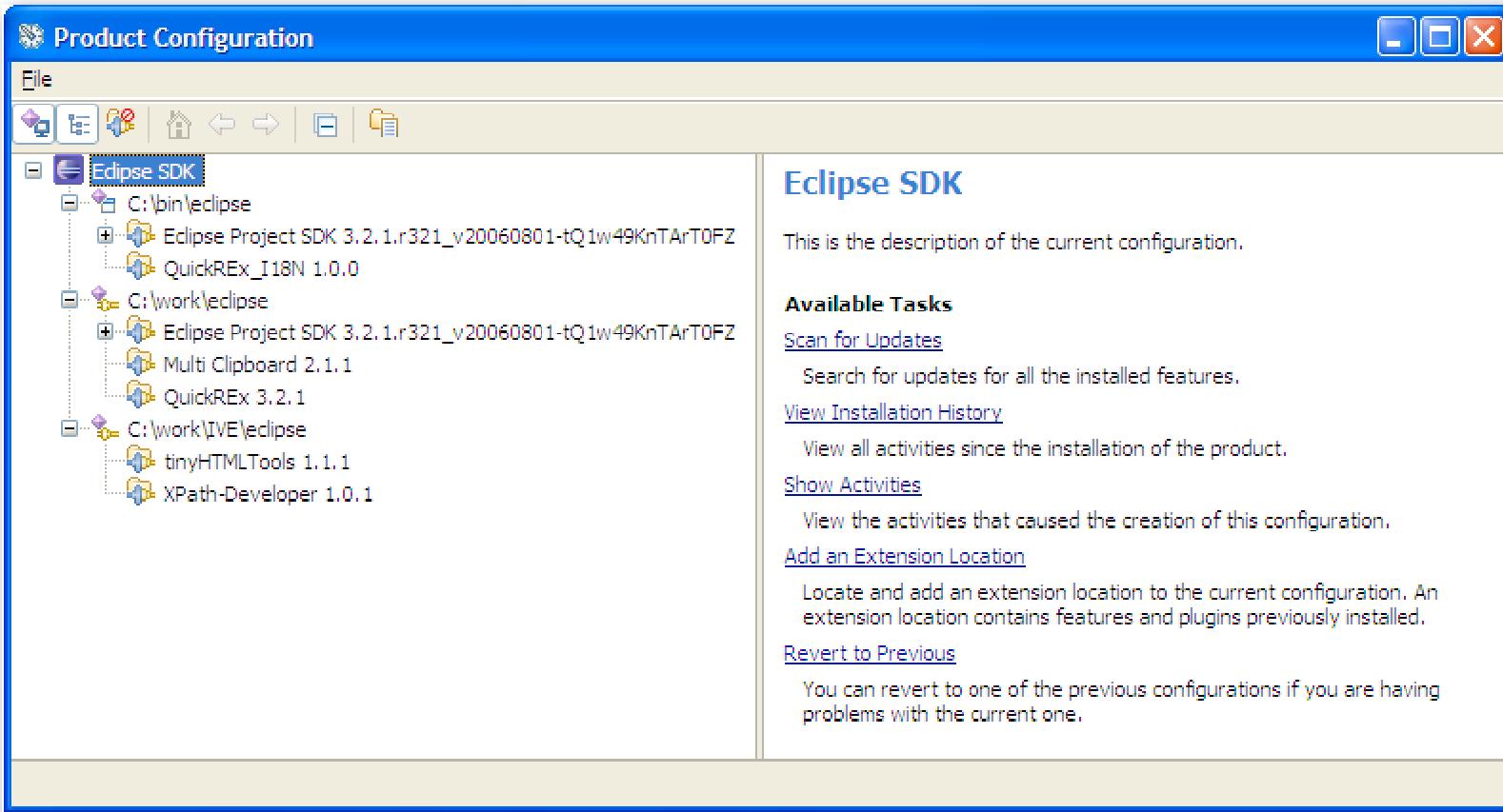
# eclipse plug-in versioning

the goal: everyone on the project has the exact same versions of the exact same plug-ins

create a new eclipse configuration

create a new empty directory named  
**.eclipse-extension**

create 2 empty directories: **features & plugins**



# eclipse plugin versioning

create a exemplar machine

create a new configuration

add plugins and features

check configuration into version control

new machine's configurations point to it

any time you find yourself  
pointing to the same thing  
more than once,  
canonicalize it with  
indirection





*DRY says that every piece of system knowledge should have one authoritative, unambiguous representation. Every piece of knowledge in the development of something should have a single representation. A system's knowledge is far broader than just its code. It refers to database schemas, test plans, the build system, even documentation.*

*the pragmatic programmer - andy hunt, dave thomas*

# dry o/r

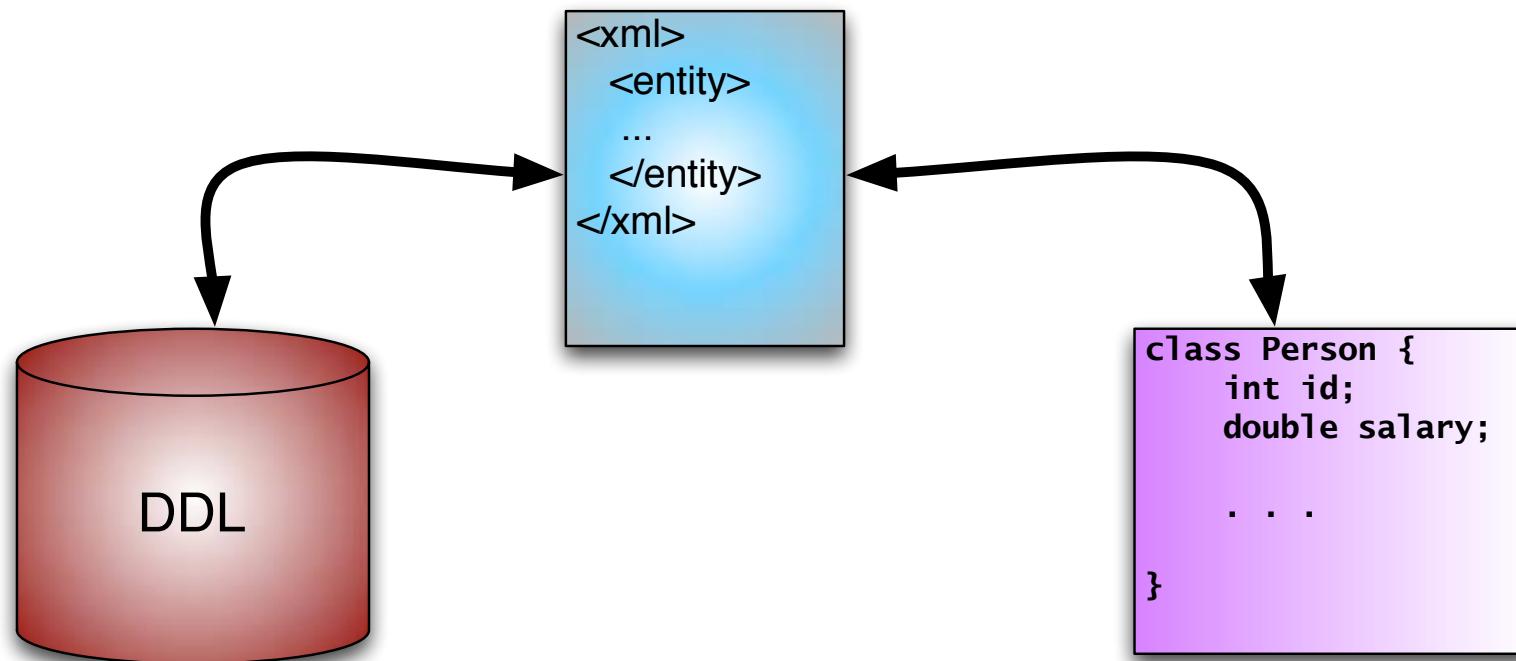
object-relational mapping is one of the most common dry violations

database schema + xml configuration + pojo > I

decide on the canonical representation

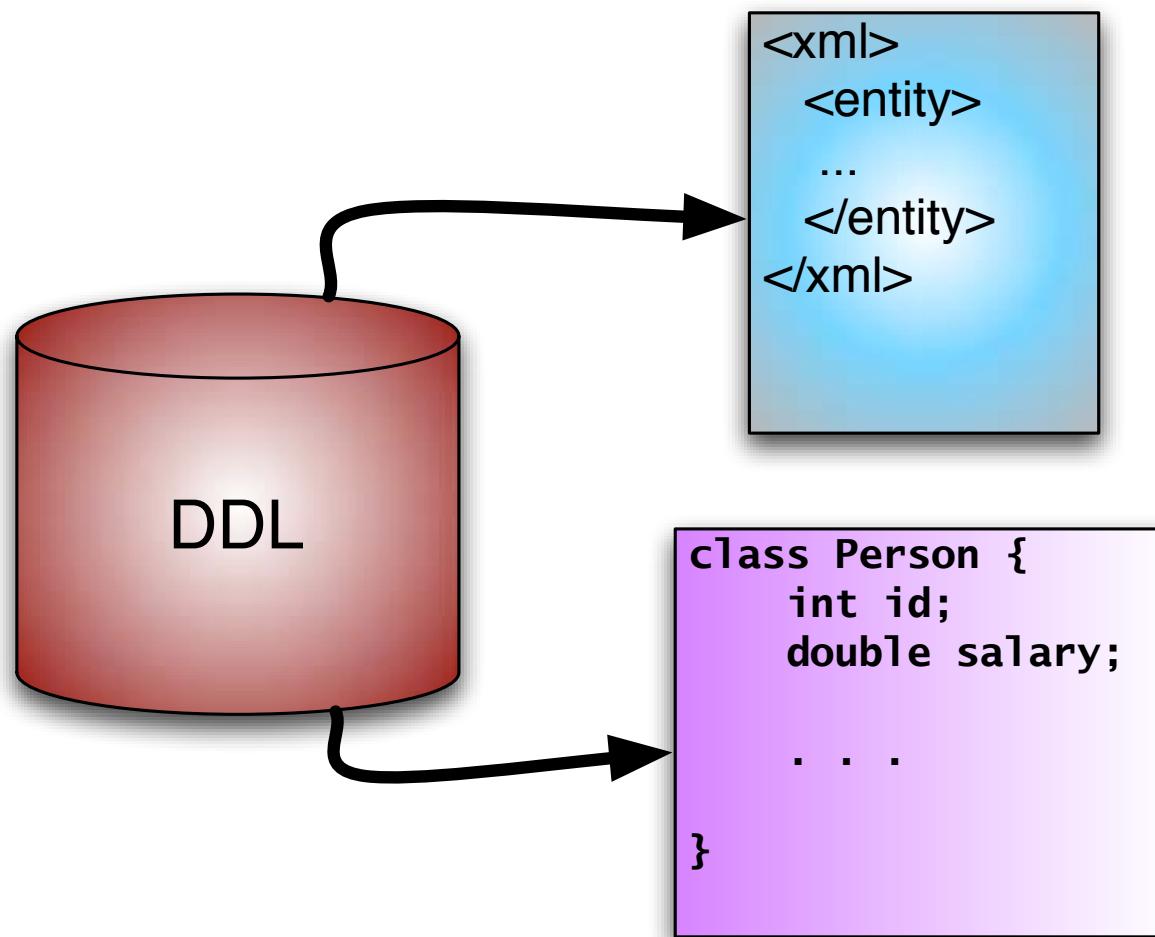
generate the others

# the scenario



where's the information?

# canonical representation



# the target

```
<sqlMap namespace='event'>
  <typeAlias type='com.nealford.conf.canonicality.Event' alias='Event' />
  <resultMap id='eventResult' class='Event'>
    <result property='description' column='DESCRIPTION' />
    <result property='eventKey' column='EVENT_KEY' />
    <result property='start' column='START' />
    <result property='eventType' column='EVENT_TYPE' />
    <result property='duration' column='DURATION' />
  </resultMap>
  <select resultMap='eventResult' id='getEvents'>select * from event where id = ?</select>
  <select resultClass='com.nealford.conf.canonicality.Event' id='getEvent'>
    select * from event where id = #value#
  </select>
</sqlMap>
```

# build event sql map

```
class GenerateEventSqlMap {  
    static final SQL =  
        ["sqlUrl":"jdbc:derby:/Users/jNf/work/derby_data/schedule",  
         "driverClass":"org.apache.derby.jdbc.EmbeddedDriver"]  
    def _file_name  
    def types = [:]  
  
    def GenerateEventSqlMap(file_name) {  
        _file_name = file_name  
    }  
}
```

```
def columnNames() {
    Class.forName(SQL["driverClass"])
    def rs = DriverManager.getConnection(SQL["sqlUrl"]).createStatement().
        executeQuery("select * from event where 1=0")

    def rsmd = rs.getMetaData()
    def columns = []
    for (index in 1..rsmd.getColumnCount()) {
        columns << rsmd.getColumnName(index)
        types.put(camelize(rsmd.getColumnName(index)),
                  rsmd.getColumnTypeName(index))
    }
    return columns
}
```

```
def camelized_columns() {
    def cc = []
    columnNames().each { c ->
        cc << camelize(c)
    }
    cc
}

def camelize(name) {
    def newName = name.toLowerCase().split("_").collect() {
        it.substring(0, 1).toUpperCase() + it.substring(1, it.length())
    }.join()
    newName.substring(0, 1).toLowerCase() +
    newName.substring(1, newName.length())
}
```

```
def columnMap() {  
    def columnMap = [:]  
    for (colName in columnNames())  
        columnMap.put(camelize(colName), colName)  
    return columnMap  
}
```

```
def create_mapping_file() {
    def writer = new StringWriter()
    def xml = new MarkupBuilder(writer)
    xml.sqlMap(namespace:'event') {
        typeAlias(alias:'Event',
                  type:'com.nealford.conf.canonicality.Event')
        resultMap(id:'eventResult', class:'Event') {
            columnMap().each() {key, value ->
                result(property:"${key}", column:"${value}")
            }
            select(id:'getEvents', resultMap:'eventResult',
                   'select * from event where id = ?')
            select(id:"getEvent",
                   resultClass:"com.nealford.conf.canonicality.Event",
                   "select * from event where id = #value#")
        }
    }
    new File(_file_name).withWriter { w ->
        w.writeLine("${writer.toString()}")
    }
}
```

# generated sql map

```
<sqlMap namespace='event'>
  <typeAlias type='com.nealford.conf.canonicality.Event' alias='Event' />
  <resultMap id='eventResult' class='Event'>
    <result property='description' column='DESCRIPTION' />
    <result property='eventKey' column='EVENT_KEY' />
    <result property='start' column='START' />
    <result property='eventType' column='EVENT_TYPE' />
    <result property='duration' column='DURATION' />
  </resultMap>
  <select resultMap='eventResult' id='getEvents'>select * from event where id = ?</select>
  <select resultClass='com.nealford.conf.canonicality.Event' id='getEvent'>
    select * from event where id = #value#
  </select>
</sqlMap>
```

# step 2: class builder

```
class ClassBuilder {  
    def imports = []  
    def fields = [:]  
    def file_name  
    def package_name  
  
    def ClassBuilder(imports, fields, file_name, package_name) {  
        this.imports = imports  
        this.fields = fields  
        this.file_name = file_name  
        this.package_name = package_name  
    }  
  
    def write_imports(w) {  
        imports.each { i ->  
            w.writeLine("import ${i};")  
        }  
        w.writeLine("")  
    }  
}
```

```
def write_classname(w) {
    def class_name_with_extension = file_name.substring(
        file_name.lastIndexOf("/") + 1, file_name.length());
    w.writeLine("public class " +
        class_name_with_extension.substring(0,
            class_name_with_extension.length() - 5) + " {")
}

def write_fields(w) {
    fields.each { name, type ->
        w.writeLine("\t$type $name;");
    }
    w.writeLine("")
}
```

```
public class Event {
    String description;
    int eventKey;
    String start;
    int eventType;
    int duration;
```

```
def write_properties(w) {
    fields.each { name, type ->
        def cap_name = name.charAt(0).toString().toUpperCase() +
            name.substring(1)
        w.writeLine("\tpublic ${type} get${cap_name}() {" +
                    "\t\treturn ${name};\n\t}\n")
        w.writeLine("\tpublic void set${cap_name}(${type} ${name}) {" +
                    "\t\tthis.${name} = ${name};\n\t}\n")
    }
}

    return description;
}

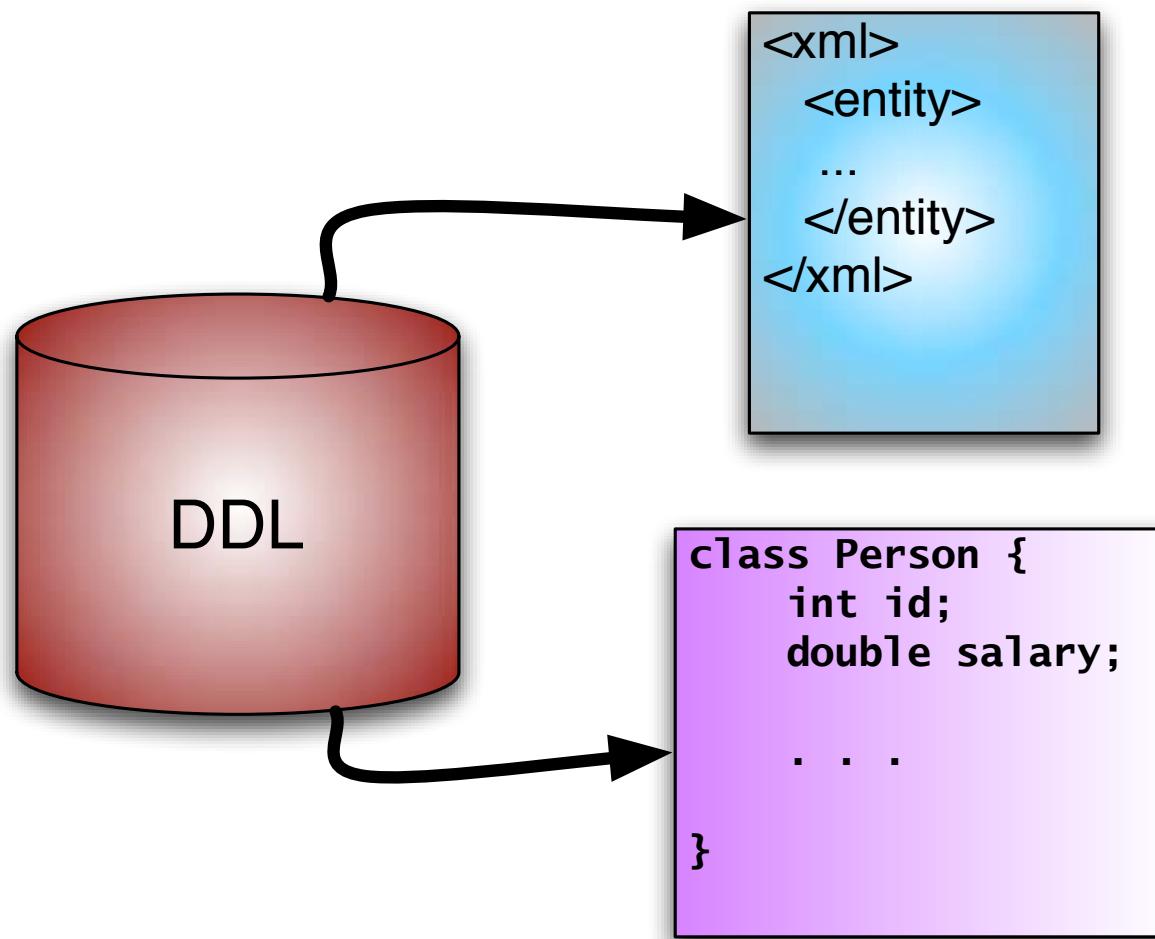
public void setDescription(String description) {
    this.description = description;
}

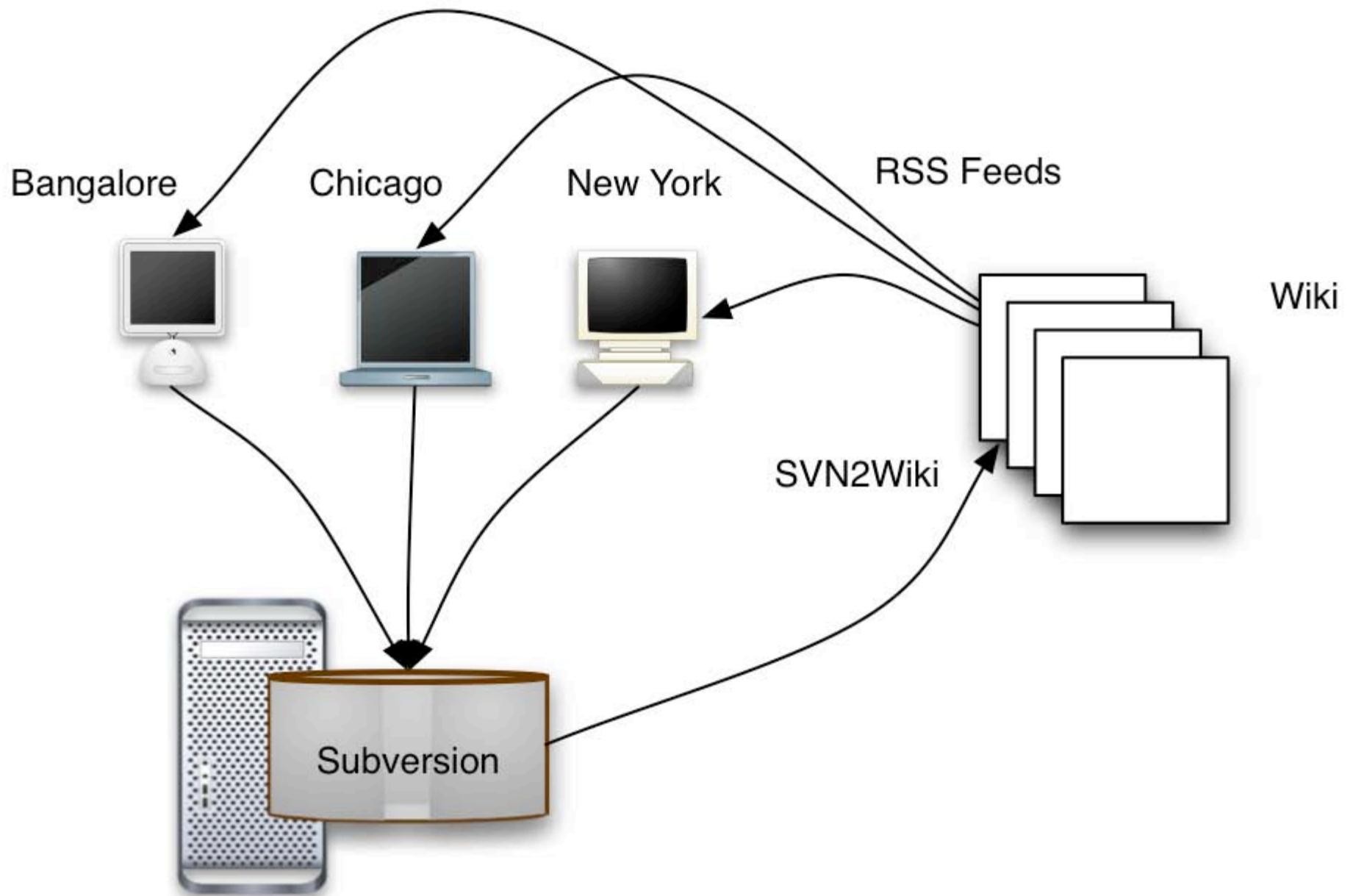
// . . .
```

```
def generate_class_file() {
    new File(file_name).withWriter { w ->
        w.writeLine("package ${package_name};\n")
        write_imports(w)
        write_classname(w)
        write_fields(w)
        write_properties(w)
        w.writeLine("}")
    }
}
```

```
public class Event {  
    String description;  
    int eventKey;  
    String start;  
    int eventType;  
    int duration;  
  
    public String getDescription() {  
        return description;  
    }  
  
    public void setDescription(String description) {  
        this.description = description;  
    }  
  
    public int getEventKey() {  
        return eventKey;  
    }  
  
    public void setEventKey(int eventKey) {
```

# canonical representation

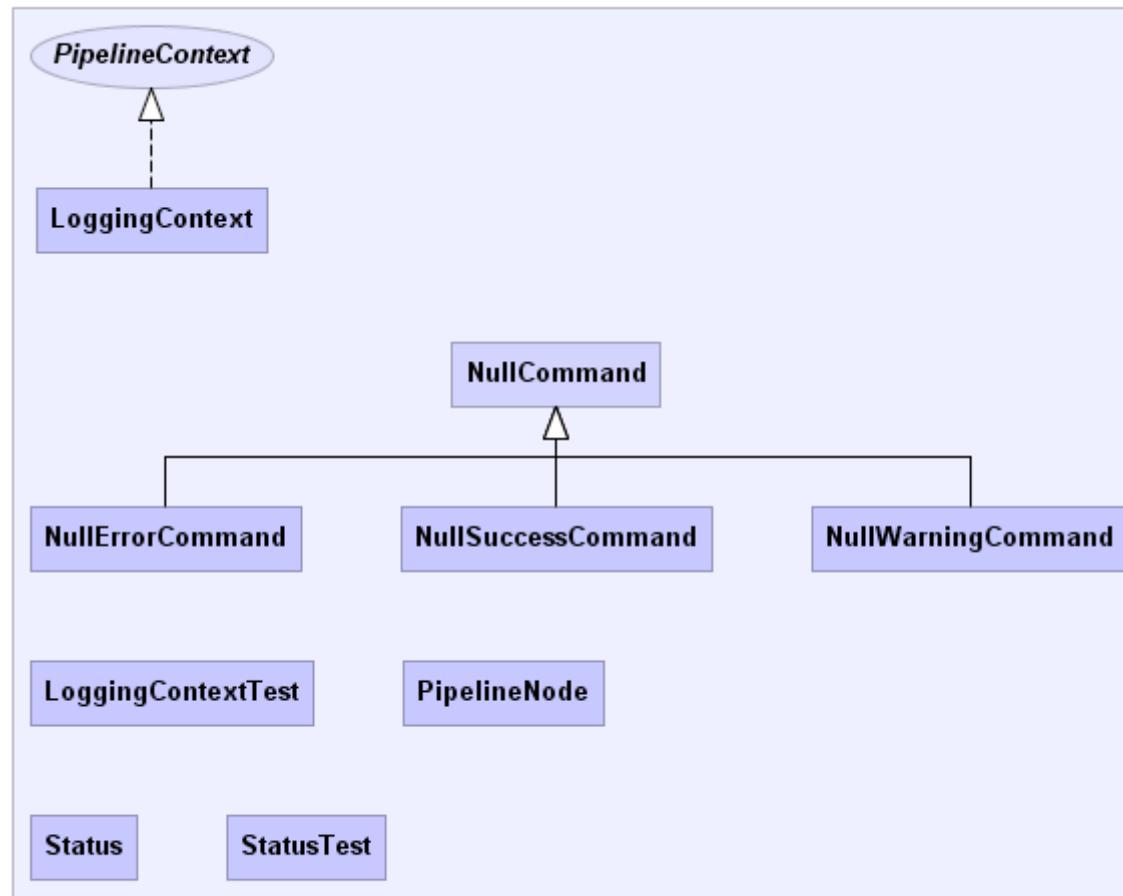




Chicago

# dry documentation

# dry diagrams



Generated by yDoc Evaluation Version

# dry schemas

the requirement: entity-relationship diagrams  
for each iteration

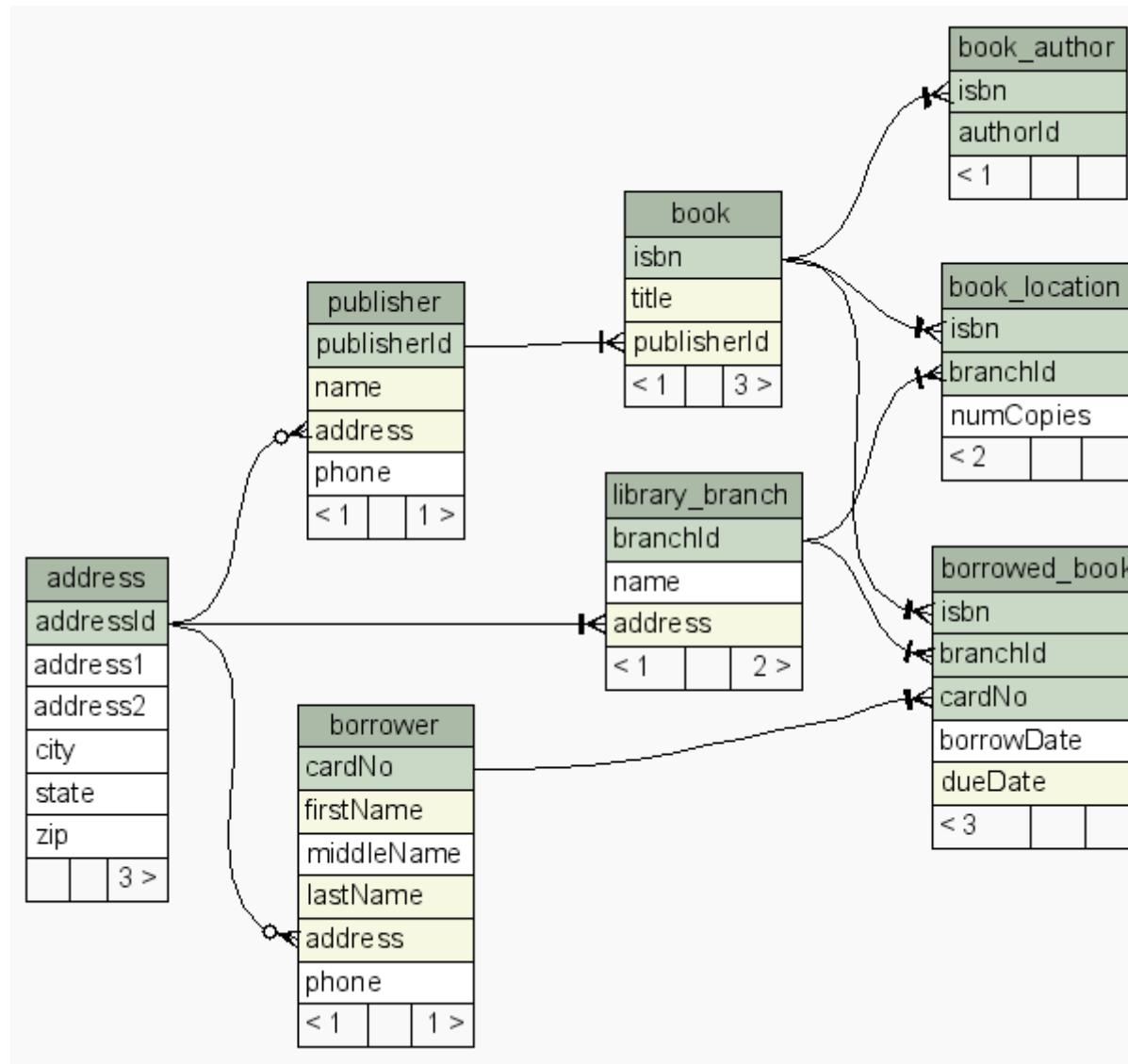
schemaspy

open source schema diagrammer

written in java

generates acceptable html

# dry schemas



# glenn vanderburg on dry

You end up rediscovering some of the more common patterns, so you actually understand them.

The experienced people on the team have plenty of opportunities to teach about various encapsulation mechanisms, design tactics, etc., because the junior people often can't figure out how to eliminate duplication and have to ask for help. (And when you explain these things in context, they tend to stick a little better.)”

# questions?

please fill out the session evaluations  
slides & samples available at [nealford.com](http://nealford.com)



This work is licensed under the Creative Commons  
Attribution-Noncommercial-Share Alike 2.5 License.

<http://creativecommons.org/licenses/by-nc-sa/2.5/>

**NEAL FORD** thoughtworker / meme wrangler

**ThoughtWorks**

14 Wall St, Suite 2019, New York, NY 10005

nford@thoughtworks.com  
[www.nealford.com](http://www.nealford.com)  
[www.thoughtworks.com](http://www.thoughtworks.com)  
[memeagora.blogspot.com](http://memeagora.blogspot.com)

# resources

Launchy

[www.launchy.net](http://www.launchy.net)

Colibri

[colibri.leetspeak.org](http://colibri.leetspeak.org)

Enzo

[www.humanized.com](http://www.humanized.com)

Quicksilver

[www.blacktree.com](http://www.blacktree.com)

CLCL

[www.nakka.com/soft/clcl](http://www.nakka.com/soft/clcl)

# resources

iClip

[inventive.us/iClip](http://inventive.us/iClip)

JumpCut

[jumpcut.sourceforge.net](http://jumpcut.sourceforge.net)

Cygwin

[www.cygwin.com](http://www.cygwin.com)

command prompt explorer bar

<http://www.codeproject.com/KB/cs/commandbar.aspx>

Auto HotKey

[www.autohotkey.com](http://www.autohotkey.com)

# resources

TextExpander

[www.smileonmymac.com/textexpander/index.html](http://www.smileonmymac.com/textexpander/index.html)

Typinator

[www.ergonis.com/products/typinator](http://www.ergonis.com/products/typinator)

BuildiX

[buildix.thoughtworks.com](http://buildix.thoughtworks.com)

Panopticode

<http://www.panopticode.org/>

DbDeploy

[dbdeploy.com](http://dbdeploy.com)

# resources

## Windows Power Toys

[www.microsoft.com/windowsxp/downloads/powertoys/xppowertoys.mspx](http://www.microsoft.com/windowsxp/downloads/powertoys/xppowertoys.mspx)

## Google Desktop Search

[desktop.google.com](http://desktop.google.com)

## Larry's "Any Text File" Indexer

[desktop.google.com/plugins/i/indexitall.html](http://desktop.google.com/plugins/i/indexitall.html)

## "metaphor shear" essay

[www.cryptonomicon.com](http://www.cryptonomicon.com)

# resources

Yahoo Pipes  
[pipes.yahoo.com](http://pipes.yahoo.com)

Selenium  
[www.openqa.org](http://www.openqa.org)

Windows Power Shell (Monad)

<http://www.microsoft.com/windowsserver2003/technologies/management/powershell/default.mspx>