

NetBeans Platform

Tom Wheeler

11/11/09

SIUE :: November 2009

What I'm Going to Cover

- What is NetBeans?
- Difference between IDE and Platform
- Why use the platform
 - ◇ Feature/benefits/disadvantages/competitors
- What do platform apps look like?
- Define some basic terms
- Overview of relevant APIs

What is NetBeans?

- An open source project
 - ◇ Sponsored by Sun Microsystems
 - ◇ 100% pure Java
 - ◇ Way better than you probably remember it
- Both an IDE *and* a platform

OK, So What is the NB Platform?

- The platform is...
 - ◇ A bunch of plug-ins (modules)
 - ◇ 100% Pure Java
 - ◇ A framework for application development
 - ◇ What's left when IDE features removed
- IDE is therefore a platform-based app.
 - ◇ IDE = Platform + IDE modules

A Platform App is a Platform

- If the platform is a foundation...
 - ◇ Which consists of a bunch of modules
 - ◇ And you can extend it by adding modules
 - ◇ And doing so creates a new application
 - ◇ And you can add modules to that new app
 - ◇ *Then isn't the new app also a platform?*

Demonstration

- If the IDE is a platform-based app...
- Shouldn't I be able to make my own IDE?

And Now for a Rhetorical Question...

- When is the last time...
 - ◇ You wrote a serious Web app in Java
 - ◇ Using only servlets and JSPs?

Why Use a Platform for Swing Apps?

- Frameworks are widespread for Web apps
- But seldom used for Swing...
 - ◇ No good reason for this, AFAIK
- 37% of any Swing app's code:
 - ◇ Is identical to 37% of any other Swing app
 - ◇ I totally made that up, but probably close

More Good Reasons to Use a Platform

- Support for modules/plugins
 - ◇ With dependency management
 - ◇ Deploy updates and new features easily
- Help your application grow over time
 - ◇ You're likely to create better APIs
 - ◇ Versioning support for smoother upgrades
- #1 Reason:
 - ◇ Spend your time on actual business logic

OK, So What Platforms Exist?

- NetBeans Platform
- Swing Application Framework (JSR-296)
- Eclipse RCP
- Spring RCP
- Countless others
 - ◇ But probably none you'd consider worthy

Swing App Framework – JSR 296

- <https://appframework.dev.java.net/>
- Supports some basic needs, like
 - ◇ Loading images / managing Actions
 - ◇ Basic data storage (like frame geometry)
- Does not support
 - ◇ Branding, modules, dependencies, windowing
- Migration plan for when you outgrow it?
- Most people consider it dead now.

NetBeans Platform

- <http://platform.netbeans.org/>
- Mature (platform available since 2001)
- Open source (CDDL + GPL)
- 100% Pure Java
 - ◇ Easily reuse Swing code
 - ◇ Uses Ant extensively
- Many features
- Adequate documentation and examples

Eclipse RCP

- <http://www.eclipse.org/rcp/>
- Mature: RCP available since late 2003 (?)
 - ◇ Open source (EPL – a CPL Variant)
 - ◇ Adequate documentation and examples
 - ◇ Many features
- Use of SWT: a dealbreaker for many!
 - ◇ Beware of platform limitations
 - ◇ Reusing existing Swing code is tough

Spring RCP

- <http://spring-rich-c.sourceforge.net/>
- Sub-project of the Spring Framework
- Don't know much about it, but
 - ◇ Data binding and validation a major feature
 - ◇ Offers at least rudimentary management
 - ◇ Plugin support unknown
 - ◇ Was dormant for a long time
 - ◇ Current version is 1.1.0, released 6/09

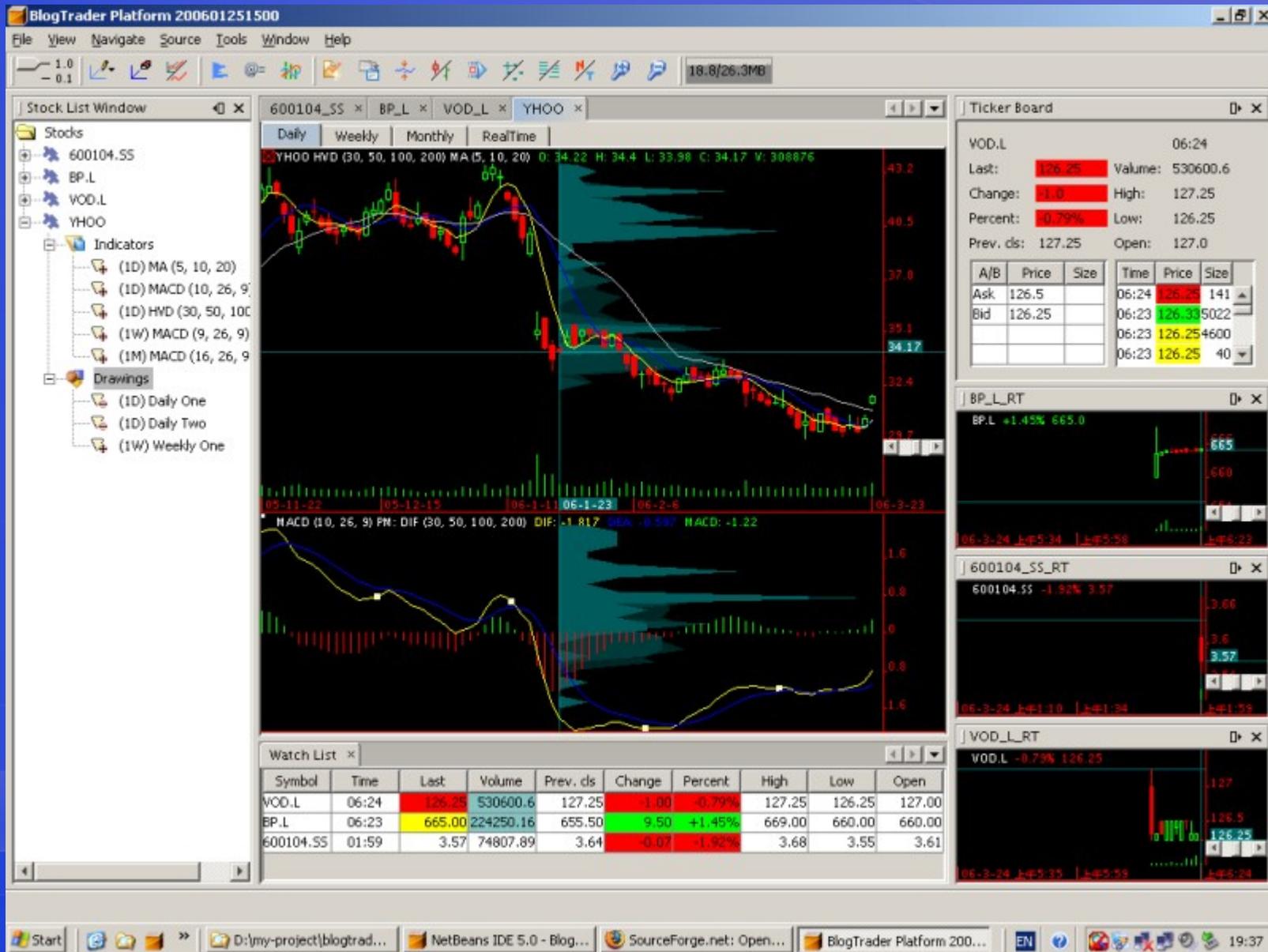
Countless Other Platforms

- There are lots of other minor players
- Some are relatively full-featured
 - ◇ But immature
- Others are relatively mature
 - ◇ But focus on a single feature
 - ◇ Example: Java Plugin Framework (JPF)
 - ◇ <http://jpf.sourceforge.net/>

OK, So Which Should I Choose?

- Typically NetBeans vs. Eclipse
 - ◇ Features are roughly equivalent
 - ◇ Both are probably good choices
 - ◇ Depends on exact requirements
 - ◇ SWT was a dealbreaker for me, but YMMV
- Also note potential IDE “lock-in”
 - ◇ Eclipse RCP *effectively requires* Eclipse IDE
 - ◇ NB Platform *heavily favors* NetBeans IDE
 - ◇ Ant integration allows other IDEs somewhat

NB Platform Example: AIOTrade

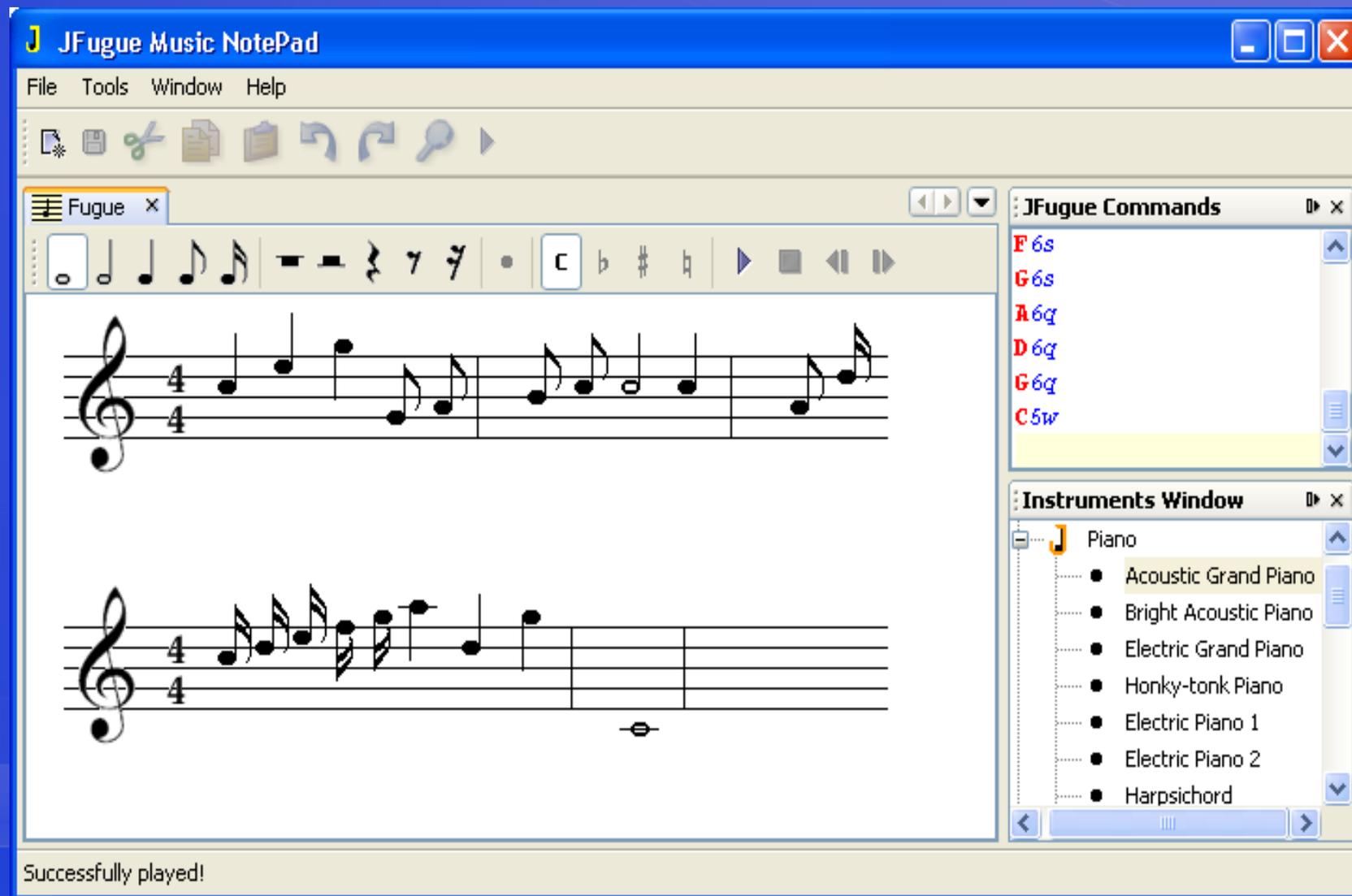


NB Platform Example: blueMarine

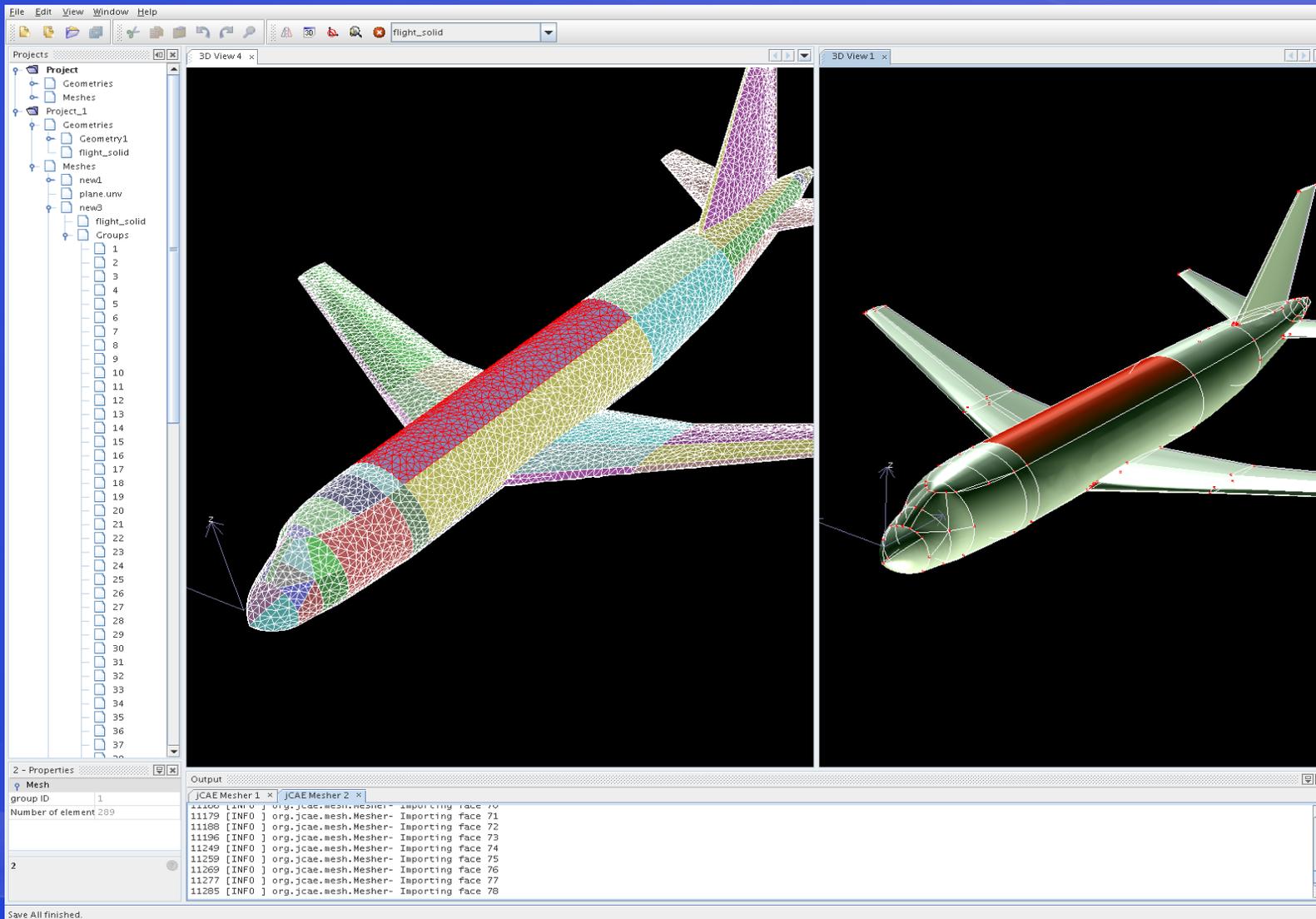
The screenshot displays the blueMarine software interface, which is a platform for managing and viewing digital content. The interface is divided into several sections:

- Categories:** A hierarchical tree view on the left side, showing taxonomic levels from kingdom to species. The 'species' level is expanded, showing a list of bird species.
- Tags:** A list of tags for the selected species, including 'Acrocephalus scirpaceus (Reed warbler)', 'Actitis hypoleucos (Common sandpiper)', and 'Anser anser (Greylag goose)'. The 'Anser anser (Greylag goose)' tag is selected.
- Video Player:** A large central window displaying a video of two Greylag geese standing in a field. The video player includes a film strip at the bottom and a clock on the right side.
- Web Browser:** A window on the right side displaying the RSPB (Royal Society for the Protection of Birds) website page for the Greylag goose. The page includes a search bar, a list of related species, a map of the UK and Europe showing the distribution of the Greylag goose, and a description of the species.

NB Platform Example: Music Notepad



NB Platform Example: JCAE



NB Platform Example: Hadoop Studio

Creating class loader...OK.
Loading main class org.apache.hadoop.examples.ExampleDriver...OK.
Inspecting main class for properties...OK.
Loading main method...OK.

Execute Hadoop Job
Runs your Hadoop job on a live cluster.

Main Class:

Target Cluster: Add...

Target Filesystem: Add...

Parameters:
(given to main())

Output Data
Reduced to 1635 records at Wed Jun 03 02:23:02 BST 2009

| Key In | Values In | Key Out | Value Out |
|--------|---------------------|---------|-----------|
| A | [1, 1, 1, 1, 1, ... | A | 12 |
| l | [1, 1, 1, 1, 1, ... | l | 85 |
| O | [1, 1, 1] | O | 3 |
| a | [1, 1, 1, 1, 1, ... | a | 105 |
| As | [1, 1, 1] | As | 3 |
| Be | [1, 1, 1, 1, 1] | Be | 5 |
| By | [1] | By | 1 |
| Go | [1, 1] | Go | 2 |

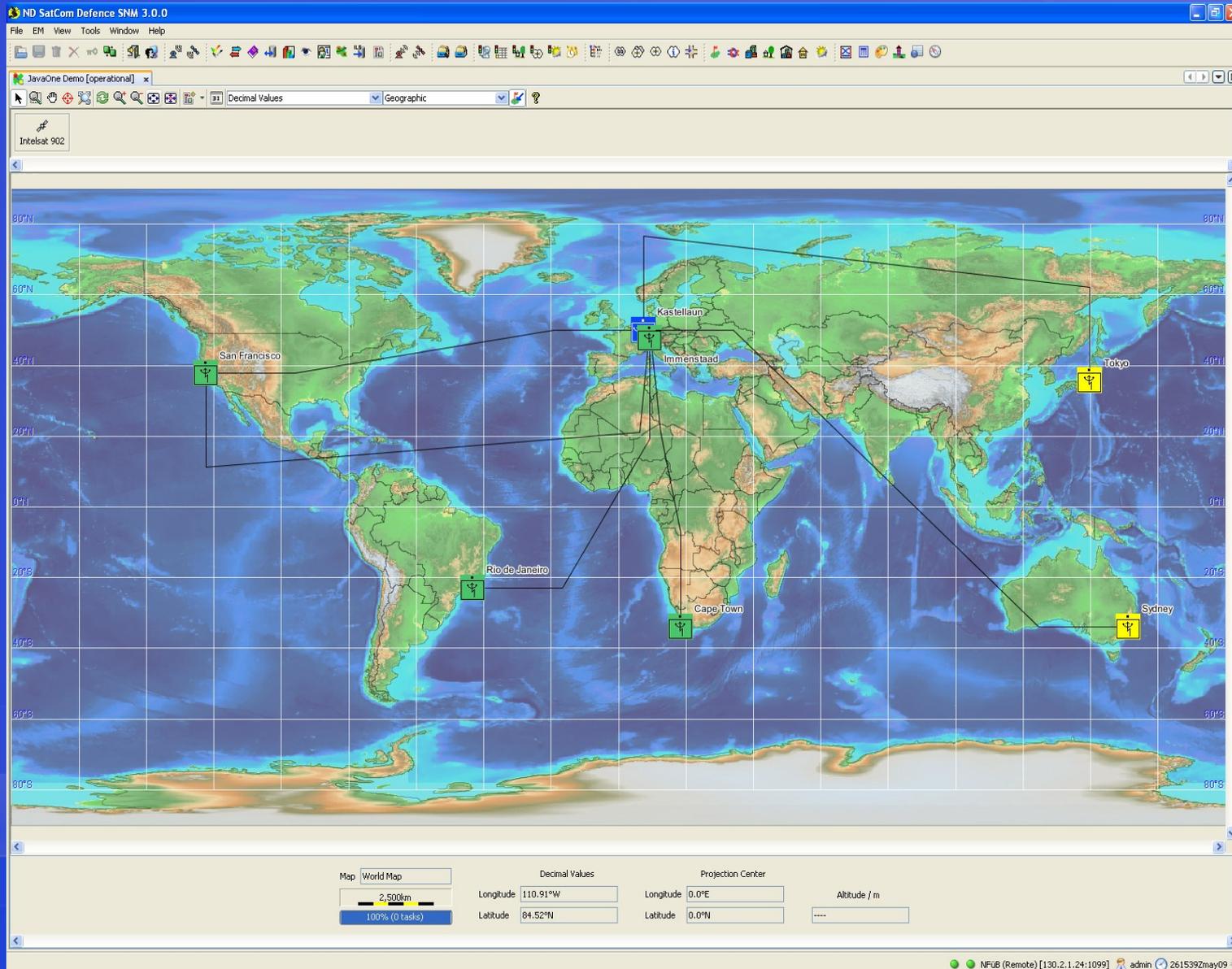
Map Attempts

Map Capacity (red line), Map Usage (blue area)

jobs

| job ID | Name | Username | State | Start Time | Map Progress | Reduce Progress | Tracking URL |
|-----------------|-------------|----------|-----------|------------------|--------------|-----------------|-----------------------|
| job_20090731... | grep-search | shevek | SUCCEEDED | Thu Aug 06 15... | 100% | 100% | http://localhost:5... |

NB Platform Example: NDSatCom



NB Platform Example: OpenMap Twitter

The screenshot displays the OpenMapTwitter application window. The title bar reads "OpenMapTwitter 200904242137". The menu bar includes "文件(E)", "编辑(E)", "视图(V)", "导航(N)", "工具(T)", "窗口(W)", and "帮助(H)". The status bar shows "51.0/74.1MB".

The main interface is divided into three sections:

- Map:** A map of San Diego, California, showing major highways (I-5, I-15, I-805, CA 52, CA 163, CA 274) and various islands (Seal Rock, Bird Rock, West Ski Island, etc.). A blue location pin is placed in the Mission Valley area.
- Twitter Friends List:** A sidebar on the right titled "Welcome" with a "Get Friend" button. It lists several Twitter accounts under the "Twitter Friends" folder:
 - allneonlike
 - San Francisco
 - Feng
 - Hangzhou, China
 - fim
 - jimijimixx
 - Chiba
 - AlexGervais
 - ReTweet_Bot
 - RSSFeedBot
 - plusplusbot
 - testuzah
 - bobtest

- Input and Output:** At the bottom, there is a text input field containing "test IIB twitter" and a "Send" button. Below the button, a status message reads "Update Successfully".

NB Platform Example: Sepix Sales Sys.

The screenshot displays the Sepix Sales System 4.0 interface. The main window title is "Sepix Sales System 4.0 200805300101". The menu bar includes "Datei", "Bearbeiten", "Ansicht", "Navigieren", "Extras", "Fenster", and "Hilfe".

Kartenfenster (Map Window):

- Buttons: Streckenabschnitte, Hinzufügen, Entfernen, Auf, Ab, Route berechnen, Kontaktassistent, Laden, Speichern.
- Streckenabschnitte list:
 - [A] Tonlandstraße 2, Hann. Münden
 - [B] Göttingen
 - [C] München
 - [D] Dresden
 - [E] Berlin
- Region anzeigen: A slider control set to approximately 10.
- Map: A map of Germany with a red route connecting points A, B, C, D, and E. A scale bar at the bottom of the map shows 0, 10, 20, and 30 units.
- Offline arbeiten:

Verwendete Ebenen (Used Layers):

- Ebenen
 - PLZ Region
 - Ausgewählte PLZ Region
 - Regionale Verteilung Adressen
 - Aktuelle Region
 - Adress-Orte
 - Berechnete Route
 - Streckenabschnitte
 - Routenzwischenpunkte
 - Blitzer
 - Mobile Blitzer
 - Ampelblitzer

Routenzwischenpunkte - Eigenschaften (Route Intermediate Points - Properties):

| | |
|------------------------------|--|
| Layer: Routenzwischenpunkte | |
| Geändert | <input type="checkbox"/> |
| Sichtbar | <input checked="" type="checkbox"/> |
| Transparenz | <input type="text"/> |
| Region: Routenzwischenpunkte | |
| Titel anzeigen | <input checked="" type="checkbox"/> |
| Textfarbe | <input type="text" value="[255,255,255]"/> |
| Linienfarbe | <input type="text" value="[204,0,102]"/> |
| Linienstärke | <input type="text" value="3.0"/> |
| Füllfarbe | <input type="text" value="[255,255,255]"/> |
| Füllfarbe aktivieren | <input type="checkbox"/> |

Transparenz (Transparency):

Steuert die Transparenz der Ebene

NB Platform Example: Sketsa SVG

The screenshot shows the Sketsa SVG Graphics Editor interface. The window title is "Sketsa SVG Graphics Editor". The menu bar includes File, Edit, View, Insert, Modify, Tools, Window, and Help. The toolbar contains icons for file operations (open, save, print) and editing (undo, redo, copy, paste, delete). The main canvas displays several SVG demos:

- Gradient Demo:** A horizontal color gradient bar and three colored circles (red, green, blue).
- Opacity Demo:** Three overlapping semi-transparent squares in red, green, and blue.
- Freeform Demo:** A line drawing of a character's head and shoulders.
- A **Yin-Yang** symbol.
- A **Dolphin** shape with a blue fill and a dashed blue selection box around it.
- A small **Spiral** graphic.

The Properties panel on the right shows settings for the selected object (the Dolphin):

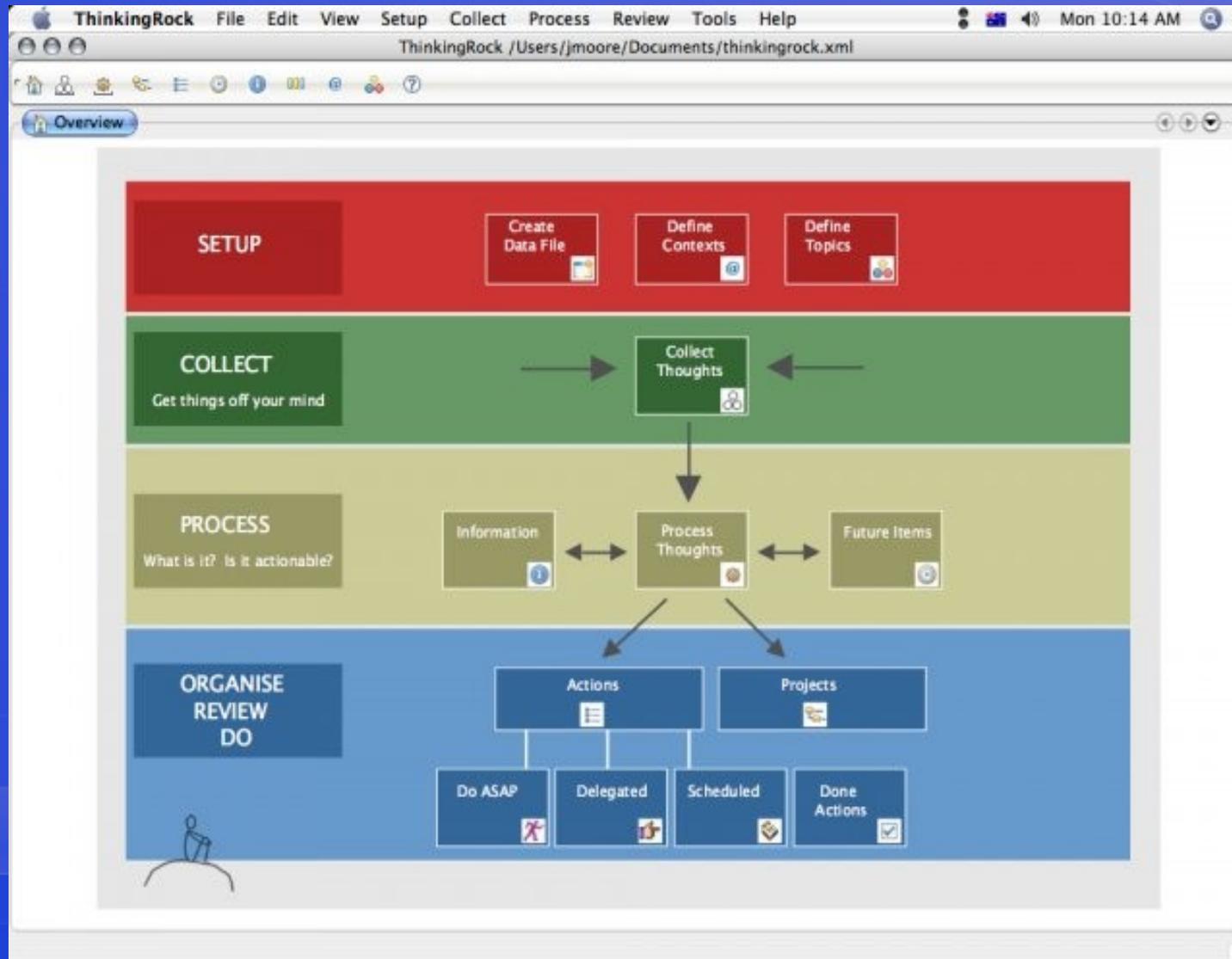
- Paint:** none (selected), color (blue), URI (empty).
- Properties:** width (1), line cap (butt), line join (miter), miter limit (4), dash array (none), dash offset (0), opacity (1).

The Defs Editor panel at the bottom right shows a list of defined elements:

| type | id |
|----------------|-----------|
| filter | feBlur |
| linearGradient | gradRGB |
| radialGradient | gradRed |
| radialGradient | gradGreen |
| radialGradient | gradBlue |

The status bar at the bottom indicates "100%" zoom and "Draw rectangular selection. Press escape to clear selection".

NB Platform Example: Thinking Rock

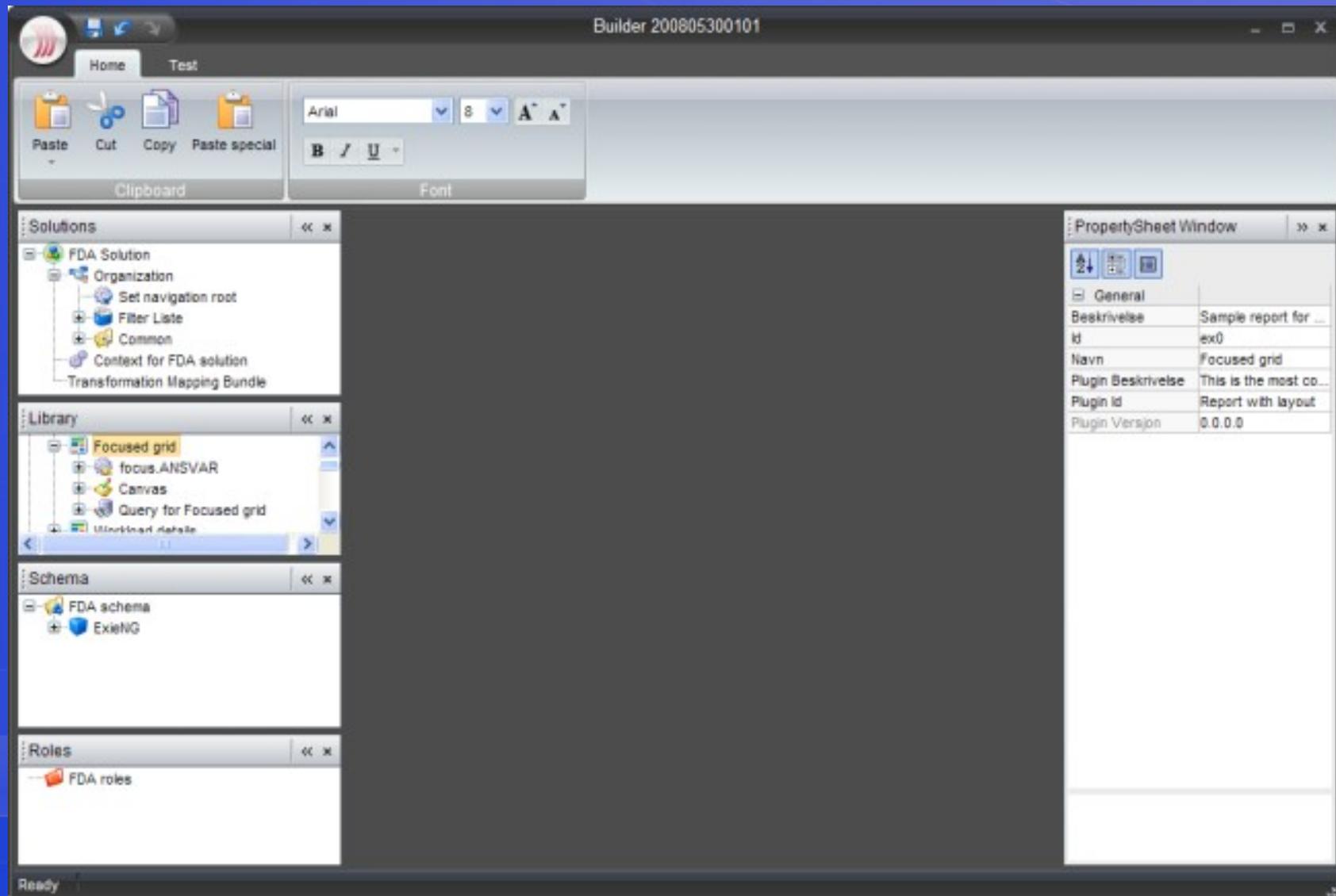


NB Platform Example: LambdaBeans

The screenshot displays the LAMBDA BEANS 1.0 Release Candidate IDE. The main window shows a code editor with Scheme code. A help window is open over the code, providing details for the `(apply procedure arg1 ... argn)` procedure. The help window includes a description: "PROC must be a procedure and ARGS must be a list. Calls PROC with the elements of the list '(append (list ARG1 ...) ARGS)' as the actual arguments." and examples: `(apply + (list 3 4)) ==> 7` and `((compose sort *) 12 75) ==> 30`. The Scheme Wiki panel at the bottom shows a list of procedures, with `apply` highlighted. The code editor shows the following Scheme code:

```
55 '(dum nil ldc (ldc 1 ld (0 . 0) eq sel (ldc 1 join) (nil ld (0 . 0) ldc 1 sub cons ld (
56 'd)))
57
58 (assert-equals 5
59 (load result /load sum ld
60
61 (0 . 0) ld (0 . 1) add rtn) ap stop)
62
63
64
65
66 (define compose
67 (lambda (f g)
68 (lambda args
69 (f (apply g args))))))
70
71
72 (apply + (list 3 4)) ==> 7
73
74
75 (define compose
76 (lambda (f g)
77 (lambda args
78 (f (apply g args))))))
79
80 ((compose sort *) 12 75) ==> 30
```

NB Platform Example: Office LAF



NetBeans Platform Features

- Modules with dependency management
- AutoUpdate Feature
- Declarative UI / branding
- Windowing system
- Integrated JavaHelp
- Deploy via JNLP (WebStart)
- Build harness (Ant scripts)

NetBeans Platform Features

- Nodes, Explorer and PropertySheet APIs
- Wizard framework
 - ◇ Supports both static and dynamic paths
- Flexible filesystem implementation
- Service Locator/Lookup
- Many handy UI utilities
- Can reuse any module from the IDE

NB Fundamentals Overview

- Suite and modules
- The layer file and the System Filesystem
- Actions
- TopComponents and Modes
- Explorer, Nodes and Properties
- Cookies and Lookups

What is a Suite?

- A suite:
 - ◇ Is a type of project in NetBeans
 - ◇ Holds configuration of a platform app.
- A suite contains
 - ◇ Ant script to build/test/run entire app
 - ◇ A list of modules
 - ◇ Branding (icons, splash screen, labels, etc.)

What is a Module?

- A module is
 - ◇ A single indivisible “piece” of an application
 - ◇ A provider of some feature or content
- A module contains
 - ◇ Exactly one manifest file and XML “layer” file
 - ◇ One or more resource bundles
 - ◇ Java code, JAR files and/or native libraries
 - ◇ Maybe some JavaHelp content

What Kinds of Modules Are There?

- There are two main types of modules
 - ◇ “Normal” (contain source code)
 - ◇ Library (contain one or more JAR files)
 - ◇ IDE wizards simplify creation of both types

What Should I Know About Visibility?

- In Java you have four types
 - ◇ private, default, protected and public
- Having “semi-public” visibility
 - ◇ For example, public only in same JAR
 - ◇ Would help in creating cleaner APIs
- NetBeans does this!
 - ◇ “Public” means “public” in that module only
 - ◇ Export the package so other modules can see

What is a FileSystem? FileObject?

- The FileSystems API in NB Platform
 - ◇ Provides access to a hierarchy of FileObjects
- Represents a “virtual” file
 - ◇ Can be on-disk like `java.io.File`
 - ◇ Can be part of an XML document
 - ◇ Can be a record in a database
 - ◇ Or something else!
- More flexible than what Java offers (now)

What is a Layer File? System FS?

- Complex apps need some type of registry
- NetBeans uses the “System Filesystem”
 - ◇ An XML-based filesystem
 - ◇ Menus, toolbars, etc. are configured here
- Each module has a “layer” file
 - ◇ This gets merged into System FS at runtime
 - ◇ Modules can add, modify and delete items
- You can use it for your storage too

What Are DataObjects?

- FileSystems provide access to FileObjects
- FileObjects provide access to raw data
- DataObjects are a level “above” this
 - ◇ Provide access to information, not just bits

And What About Actions?

- Same as in Swing, they “do things”
- Can generally use Swing `AbstractAction`
 - ◇ There are also NB-specific types
 - ◇ For both context-sensitive and stateless use
 - ◇ Makes handling `isEnabled()` easy

TopComponents a la Mode

- TopComponent is basically a JPanel
 - ◇ But also a window in NB windowing system
- Every TopComponent “lives” somewhere
 - ◇ This place is called a *mode*
- Modes are named after IDE components
 - ◇ Explorer
 - ◇ Editor
 - ◇ Output

Explorer, Nodes and Properties

- Nodes are central to NB programming
 - ◇ Presentation layer
 - ◇ Represent some type of data
 - ◇ For example: Customer, Order or Product
- Nodes are displayed in an explorer view
 - ◇ Typically a tree-based view
 - ◇ But there are other views (table, list, menu)
 - ◇ Can typically switch views w/o model change
 - ◇ Try that with Swing!

Cookies (Capabilities)

- Cookies aren't what you think
 - ◇ Have nothing to do with HTTP or X-Windows
- Represent some capability of an object
- Can dynamically add and remove them
 - ◇ For example, `SaveCookie` interface
 - ◇ Has one method: `save()`
 - ◇ When active node has a `SaveCookie`
 - ◇ File -> Save is enabled
 - ◇ Otherwise it is not

Lookups: Even Better Than a Cookie

- Lookups are a more modern version
 - ◇ Don't require you to impl. marker interface
- There is also a “Global Lookup”
 - ◇ You can code to an interface
 - ◇ Find implementation at runtime
 - ◇ Ideal for plugging in algorithms
 - ◇ Similar idea now in Java 6 (ServiceLoader)

Putting It All Together: An Example

- The StringMaster 3000!

Review Questions

- Name an application built on NB Platform
- What is a module?
- What is a suite?
- How does visibility in a NB Platform app differ from that in a plain Java app?
- Name three features of NB Platform

Recap

- NB Platform is
 - ◇ A free, open source, modular app framework
 - ◇ Built on Java and Swing
 - ◇ A better way to build large Swing apps
 - ◇ Proven technology used by many orgs
- An application is made up of
 - ◇ Exactly one suite
 - ◇ One or more modules

Exercise (30 minutes)

- I'll demonstrate how to create a simple "Hello World" NetBeans Platform app
- Then you try it on your own
- Then we'll examine the structure