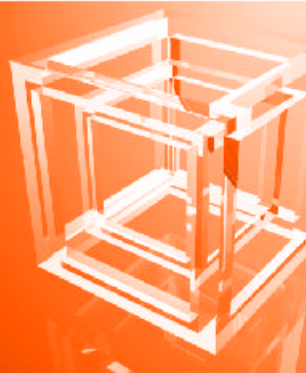


Building Applications on the NetBeans Platform

Tim Boudreau
Staff Engineer
Sun Microsystems



Bangkok, Thailand – January 2006

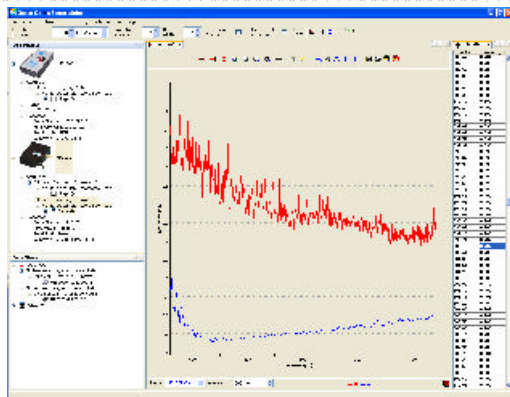
What is the NetBeans Platform?

- ? Generic desktop application
- ? Provides services almost all desktop application need
- ? Lets you concentrate on your business logic
- ? Saves years of development time
- ? Does for the desktop what app servers do for the server



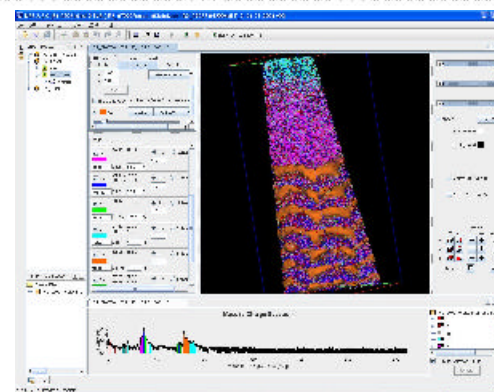
Bangkok, Thailand – January 2006

Examples – SpectraSuite



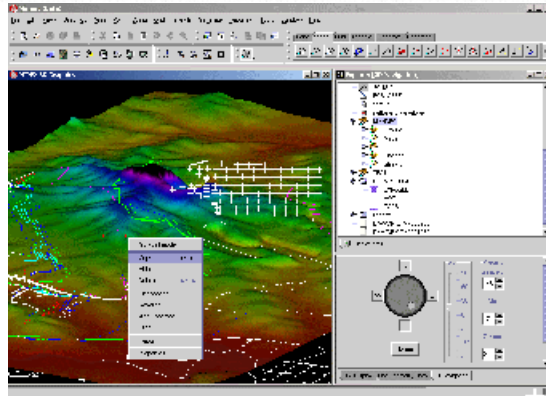
Bangkok, Thailand – January 2006

Examples – IVAS



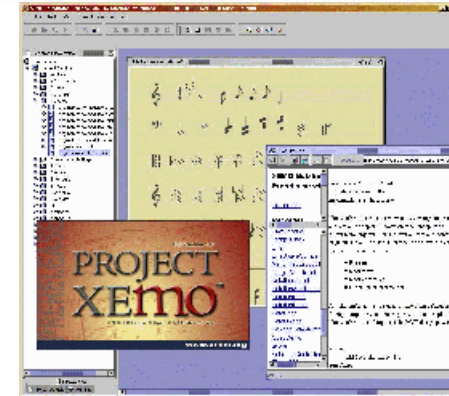
Bangkok, Thailand – January 2006

Examples – Minex



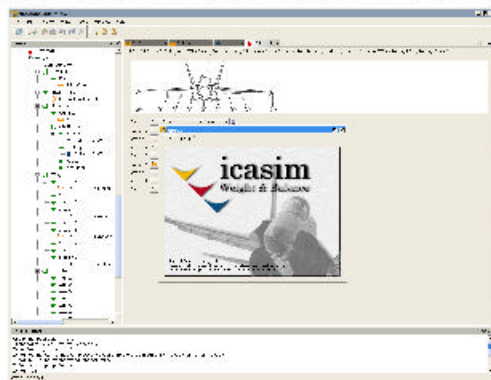
Bangkok, Thailand – January 2006

Examples – XEMO



Bangkok, Thailand – January 2006

Examples – Icasim



Bangkok, Thailand – January 2006

What does a Rich Client Platform do for you?

- ? Saves **years** of development time
 - Menus, actions, settings, windowing – no fun and easy to have bugs
 - ? The platform gives you all of this
 - ? 8 years of development – it's mature and well tested
- ? Gives you **free stuff**
- ? Gives you **better software**
 - Modularity helps maintainability



Bangkok, Thailand – January 2006

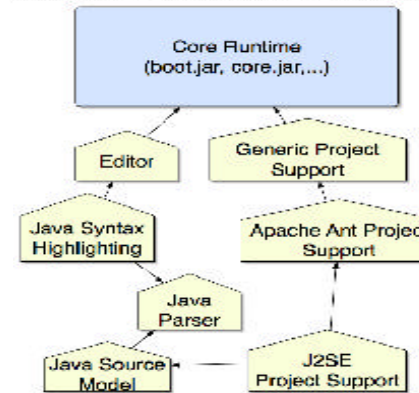
NetBeans Modules and Platform

- ? Platform == “shell”
- ? Platform + IDE modules == NetBeans IDE
- ? Platform + *your* modules == Your Application



Bangkok, Thailand – January 2006

Everything is a Module (Plug-in)



Bangkok, Thailand – January 2006

Modular does not have to *look* modular!

- ? A little UI design...
- ? And a little thought...
- ? And some good APIs...

An application composed of pieces does not have to look like a pile of spare parts



Bangkok, Thailand – January 2006

5.0 - Plugin Development Support

- ? Plug-in from a template
- ? Plug-in “suites”
- ? Build/run/debug within the IDE
- ? Hot-redeploy plugins
- ? Debug “fix and continue” (hotswap) for plug-ins
- ? Configuration GUI
- ? Branding
- ? JNLP (Java WebStart)
- ? NetBeans API class templates



Bangkok, Thailand – January 2006

Demos

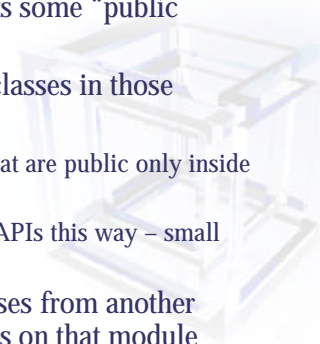
- ? A very simple application
- ? Creating a plug-in



Bangkok, Thailand – January 2006

ClassLoader Partitioning

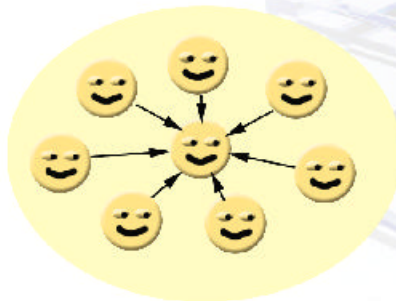
- ? A module that has an API lists some “public packages”
- ? Other modules can only see classes in those packages
 - You can have public classes that are public only inside their own JAR
 - You can design *much* cleaner APIs this way – small packages
- ? If a module wants to use classes from another module, it must say it depends on that module



Bangkok, Thailand – January 2006

Why modular applications?

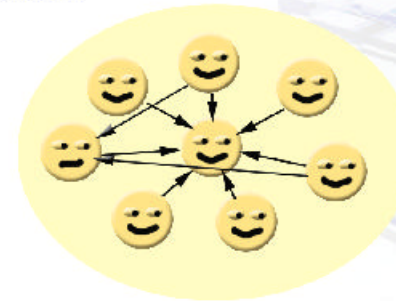
- Version 1.0 is cleanly designed



Bangkok, Thailand – January 2006

Why modular applications?

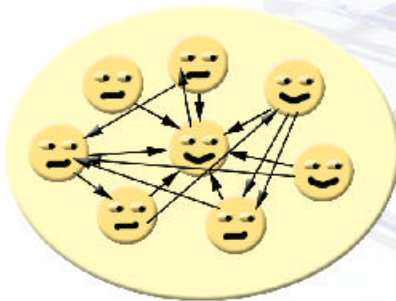
- Version 1.1...a few expedient hacks...we'll clean those up in 2.0



Bangkok, Thailand – January 2006

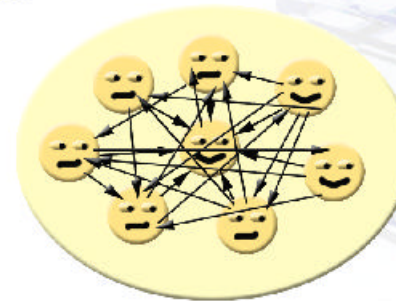
Why modular applications?

- Version 2.0...oops...but it works!



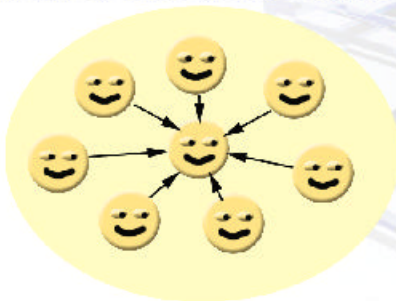
Why modular applications?

- Version 3.0...Help! Whenever I fix one bug, I create two more!



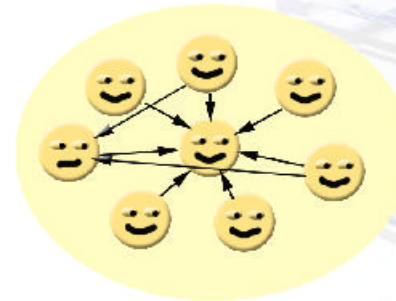
Why modular applications?

- Version 4.0 is cleanly designed. It's a complete rewrite. It was a year late, but it works...



Why modular applications?

- Version 4.1...does this look familiar?...



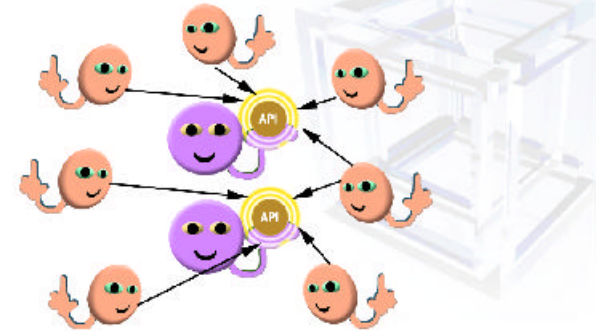
Why modular applications?

- TO BE CONTINUED...



Why modular applications?

- Healthy APIs make Healthy Applications!



Example – an extensible application

DEMO



What is a plug-in (module)

- ? JAR file with some special manifest entries

OpenIDE-Module: org.netbeans.modules.project.ant/1

OpenIDE-Module-Specification-Version: 1.6

OpenIDE-Module-Implementation-Version: 050219

OpenIDE-Module-Public-Packages: org.netbeans.api.project.ant.*, org.netbeans.spi.project.ant.*

OpenIDE-Module-Module-Dependencies: org.netbeans.modules.projectapi/1, org.netbeans.modules.queries/1, org.openide.loaders

OpenIDE-Module-Localizing-Bundle: org.netbeans/modules/project/ant/Bundle.properties

What is a plug-in (module)

? Module code-name

OpenIDE-Module: org.netbeans.modules.project.ant/1

OpenIDE-Module-Specification-Version: 1.6

OpenIDE-Module-Implementation-Version: 050219

OpenIDE-Module-Public-Packages: org.netbeans.api.project.ant.*, org.netbeans.spi.project.ant.*

OpenIDE-Module-Module-Dependencies: org.netbeans.modules.projectapi/1, org.netbeans.modules.queries/1, org.openide.loaders

OpenIDE-Module-Localizing-Bundle: org.netbeans/modules/project/ant/Bundle.properties



Bangkok, Thailand – January 2006

What is a plug-in (module)

? Modules have version numbers

OpenIDE-Module: org.netbeans.modules.project.ant/1

OpenIDE-Module-Specification-Version: 1.6

OpenIDE-Module-Implementation-Version: 050219

OpenIDE-Module-Public-Packages: org.netbeans.api.project.ant.*, org.netbeans.spi.project.ant.*

OpenIDE-Module-Module-Dependencies: org.netbeans.modules.projectapi/1, org.netbeans.modules.queries/1, org.openide.loaders

OpenIDE-Module-Localizing-Bundle: org.netbeans/modules/project/ant/Bundle.properties



Bangkok, Thailand – January 2006

What is a plug-in (module)

? Modules can expose APIs

OpenIDE-Module: org.netbeans.modules.project.ant/1

OpenIDE-Module-Specification-Version: 1.6

OpenIDE-Module-Implementation-Version: 050219

OpenIDE-Module-Public-Packages: org.netbeans.api.project.ant.*, org.netbeans.spi.project.ant.*

OpenIDE-Module-Module-Dependencies: org.netbeans.modules.projectapi/1, org.netbeans.modules.queries/1, org.openide.loaders

OpenIDE-Module-Localizing-Bundle: org.netbeans/modules/project/ant/Bundle.properties



Bangkok, Thailand – January 2006

What is a plug-in (module)

? Modules can use APIs from other modules

OpenIDE-Module: org.netbeans.modules.project.ant/1

OpenIDE-Module-Specification-Version: 1.6

OpenIDE-Module-Implementation-Version: 050219

OpenIDE-Module-Public-Packages: org.netbeans.api.project.ant.*, org.netbeans.spi.project.ant.*

OpenIDE-Module-Module-Dependencies: org.netbeans.modules.projectapi/1, org.netbeans.modules.queries/1, org.openide.loaders

OpenIDE-Module-Localizing-Bundle: org.netbeans/modules/project/ant/Bundle.properties



Bangkok, Thailand – January 2006

What is a plug-in (module)

? Everything is localized

OpenIDE-Module: org.netbeans.modules.project.ant/1

OpenIDE-Module-Specification-Version: 1.6

OpenIDE-Module-Implementation-Version: 050219

OpenIDE-Module-Public-Packages: org.netbeans.api.project.ant.*, org.netbeans.spi.project.ant.*

OpenIDE-Module-Dependencies: org.netbeans.modules.projectapi/1, org.netbeans.modules.queries/1, org.openide.loaders

OpenIDE-Module-Localizing-Bundle: org.netbeans/modules/project/ant/Bundle.properties



Bangkok, Thailand - January 2006

Demos

? Integrating a Hex Editor



Bangkok, Thailand - January 2006

Discovery - how plugins install things

- ? You don't want to run code on startup, it slows things down
- ? There are declarative ways to install things in NetBeans
 - *System Filesystem* - a virtual filesystem, XML - you put "files" that represent java classes (MyClass.instance) in "folders" that have different functions
 - *Java Extension Mechanism* - a file in the JAR's META-INF/Services directory names the interface, its content names the implementation



Bangkok, Thailand - January 2006

Resources

- ? <http://www.netbeans.org> - Main NetBeans web site
- ? <http://platform.netbeans.org> - Plugin developer resources
- ? Plugin development mailing list:
 - dev@openide.netbeans.org - subscribe on the web site, or
 - [news://news.gmane.org](http://news.gmane.org) - mirrors the mailing lists



Bangkok, Thailand - January 2006

Lookup API

- ? A way to find objects (dependency injection)
- ? Global singletons or services
 - `Lookup.getDefault().lookup(SomeInterface.class)`
- ? Objects belonging to another object
 - `node.getLookup().lookup(SomeInterface.class)`
 - `dataObject.getCooki e(SomeInterface.class)`
- ? Why lookup?
 - ? Lazy instantiation – create on demand
 - ? Evolution/Decoupling
 - ? Its hard to add/remove methods
 - ? Its easy to change what lives in an object's Lookup



Bangkok, Thailand – January 2006

Lookup Patterns

- ? “Glue” between disparate APIs
 - `Lookup.getDefault().lookup(MyService.class);`
- ? Common pattern: To ask an object for an implementation of some interface
 - `SomeObject.getLookup().lookup(SomeInterface.class);`
 - `SomeObject.getCooki e(SomeInterface.class);`



Bangkok, Thailand – January 2006

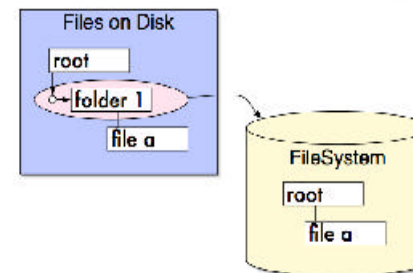
Elements of a GUI application - selection

- ? Each “tab” (TopComponent) has an *activated Node*
 - which represents something in a model
 - has a display name, actions, etc. - human readable things
 - has a *Lookup* (a Map where the keys are .class objects)
- ? There is a global Lookup which changes as focus changes:
 - `Utilities.actionsGlobalContext.lookup(OpenCooki e.class);`



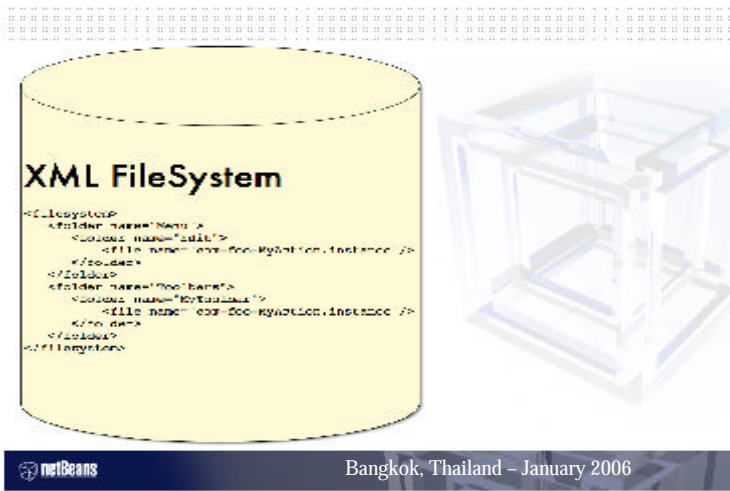
Bangkok, Thailand – January 2006

Filesystems – virtual files

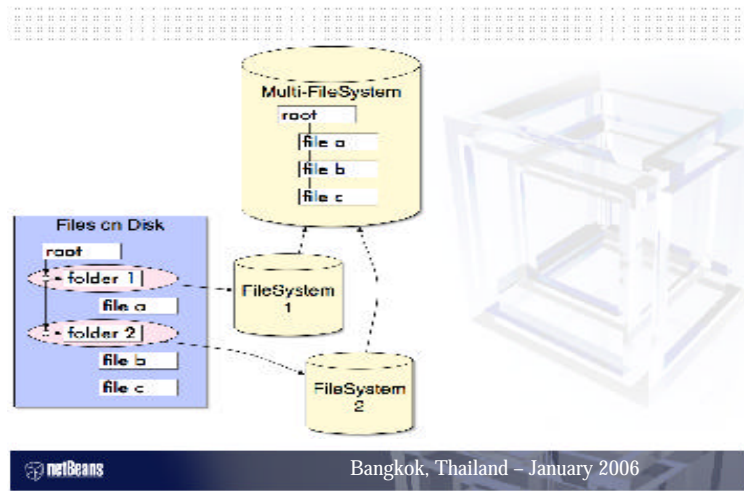


Bangkok, Thailand – January 2006

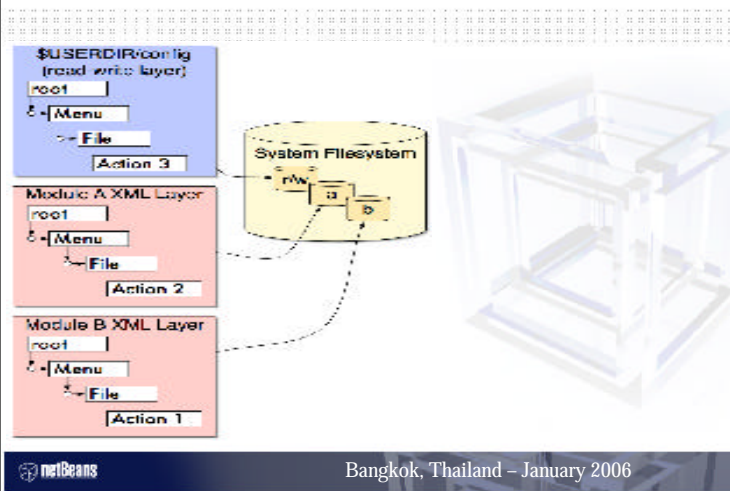
XML Filesystems – very virtual files



Layered Filesystem

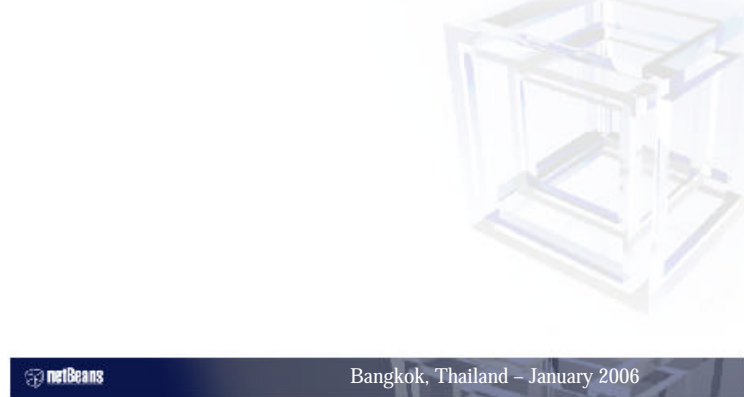


System (configuration) Filesystem



Manipulating the Platform

? Demo: Hiding the Help Menu



Modules create their own extension points

- ? System Filesystem
 - Define a folder in the System Filesystem
 - (Optionally) define an interface for what goes there
 - Let other modules add files there
- ? Global Lookup
 - Define an interface
 - Let modules register interfaces in META-INF/services



Bangkok, Thailand - January 2006

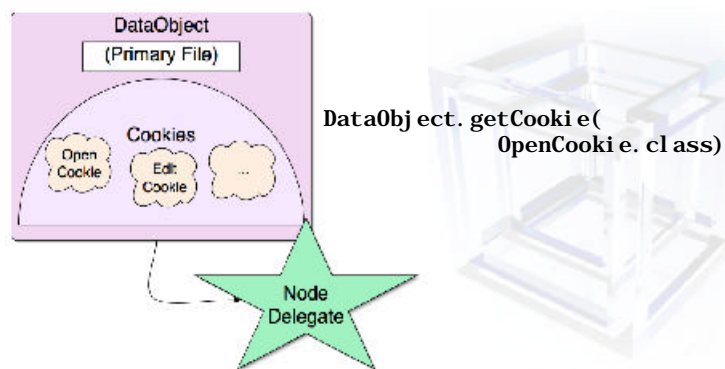
A Tour of the System Filesystem

DEMO: Browsing the System Filesystem



Bangkok, Thailand - January 2006

DataObjects wrap FileObjects



Bangkok, Thailand - January 2006

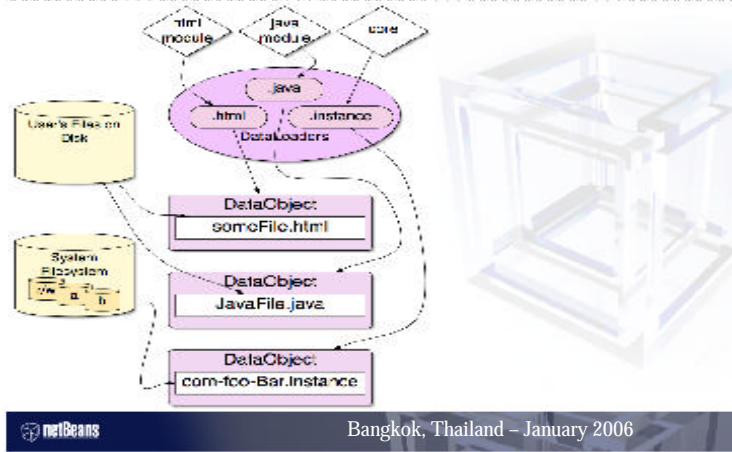
DataSystems API

- ? DataObjects – wrap 1 or more FileObjects
 - “Typed” files – knows what the content is
 - ? DataObject subclass :: MIME type
 - Programmatic access to file contents
 - ? Structural or parsed representation of contents
 - ? Support for editing, etc.
- ? DataLoaders
 - Factories for DataObject
 - Recognize and claim FileObjects by extension or contents



Bangkok, Thailand - January 2006

DataSystems



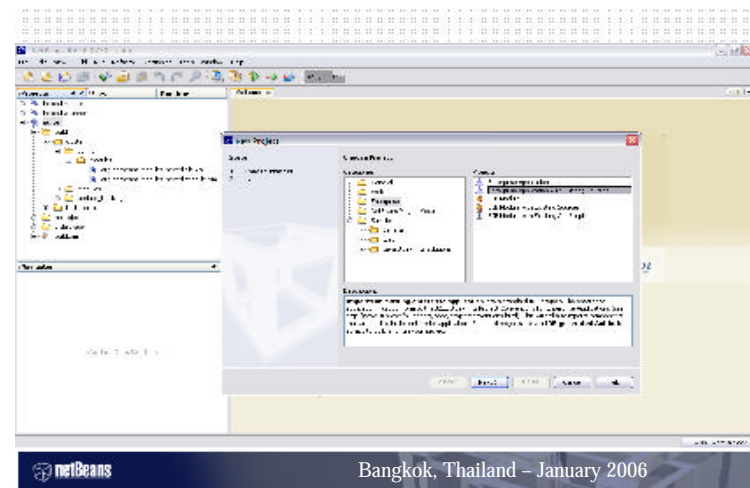
What is a Node

- ? A node has
- Human readable display name
 - Icon
 - Actions
 - Cookies (objects that can be requested from its Lookup)

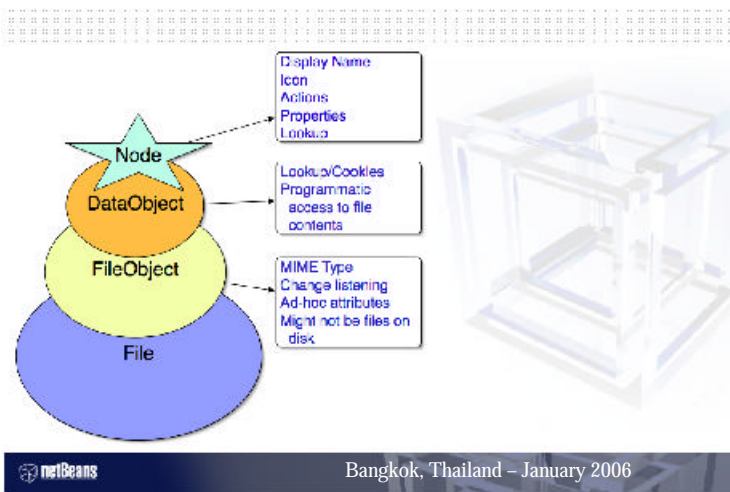
Nodes API

- ? Presentation layer on top of DataObjects
- ? Generic hierarchy API
- ? Displayed to user with the Explorer API
- Tree, List, Combo, Menu, TreeTable and other views
- ? Don't have to represent files
- ? Global selection context
- Each tab has an "activated node"

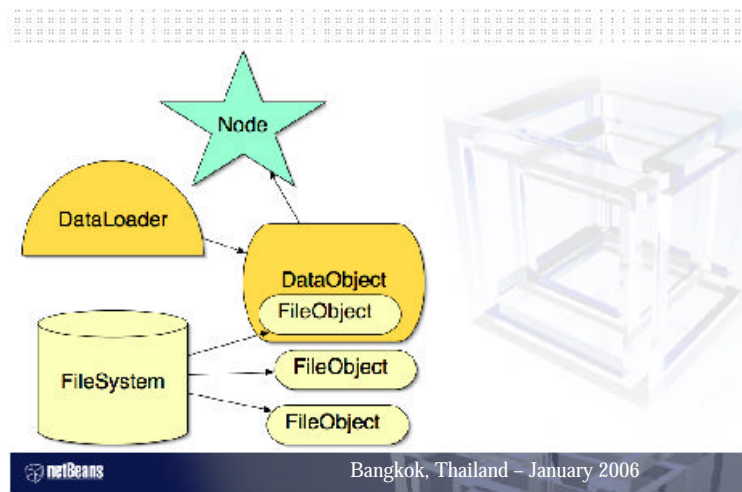
These are Nodes



From Files to Nodes



Factory Relationships – Files to Nodes



Global Lookup Registration

? Instance file in system filesystem:

```
<folder name="services">
  <file name="org-foo-mymodule-MyService.instance"/>
</folder>
```

? META-INF/services entry in jar:

? A single file that names the *interface* implemented

META-INF/services/org.openide.awt.StatusDisplayer

- With a single line of text naming the *implementation class*

com.mycom.mymodule.MyStatusDisplayer

netBeans

Bangkok, Thailand – January 2006

Global Lookup Registration

? StatusDisplayer.getDefault() does this:

```
public static StatusDisplayer getDefault() {
    return Lookup.getDefault().lookup(
        StatusDisplayer.class);
}
```

...and finds

com.mycom.mymodule.MyStatusDisplayer

netBeans

Bangkok, Thailand – January 2006

It's like Jini

- ? Don't think "I want to listen to that object"
- ? Think: "I want to watch this space and..."
 - See what's in it
 - React when objects appear or disappear
- ? `Lookup.getDefault()` is the global space
- ? Any object can own its own space – `someObject.getLookup()`...
- ? and many do (Nodes, Projects, ...)



Bangkok, Thailand – January 2006

What's in a Node's Lookup?

- ? Everything that's in its `DataObject`'s lookup
- ? Objects that implement known interfaces, such as
 - `OpenCookie` to open a file
 - Abstract syntax tree of a Java file
 - DOM tree of an XML file
 - `SaveCookie` if the file is modified
 - Whatever you want to put there...



Bangkok, Thailand – January 2006

`getLookup()` vs. `getCookie()`

- ? Newer idiom
 - ? `someObject.getLookup().lookup(SomeClass.class)`
- ? Older idiom
 - ? `someObject.getCookie(SomeClass.class)`
- ? Principle difference
 - ? Lookup is more generic
 - ? Objects in a Lookup don't have to implement the marker interface `Node.Cookie`



Bangkok, Thailand – January 2006

Things to remember

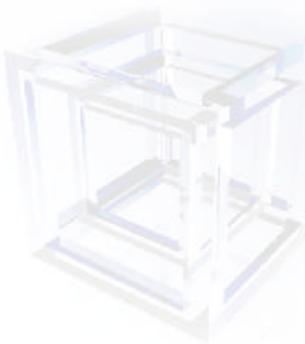
- ? Many things have two purposes
 - `FileSystems/DataSystems`
 - ? `FileObjects/DataObjects` represent the user's files
 - ? `FileObjects/DataObjects` represent configuration data
 - Lookup
 - ? Global services, singletons
 - ? Objects belonging to some other object
- ? The system filesystem is just another filesystem



Bangkok, Thailand – January 2006

Questions and Answers

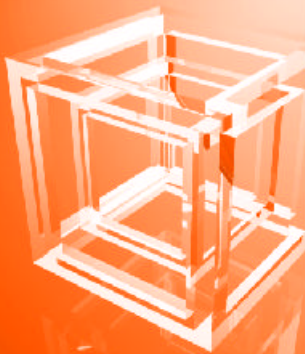
Q&A



Bangkok, Thailand – January 2006

Productivity with NetBeans, Matisse and the NetBeans Platform

Tim Boudreau
Staff Engineer
Sun Microsystems



Sao Paulo, Brazil – Nov. 2005