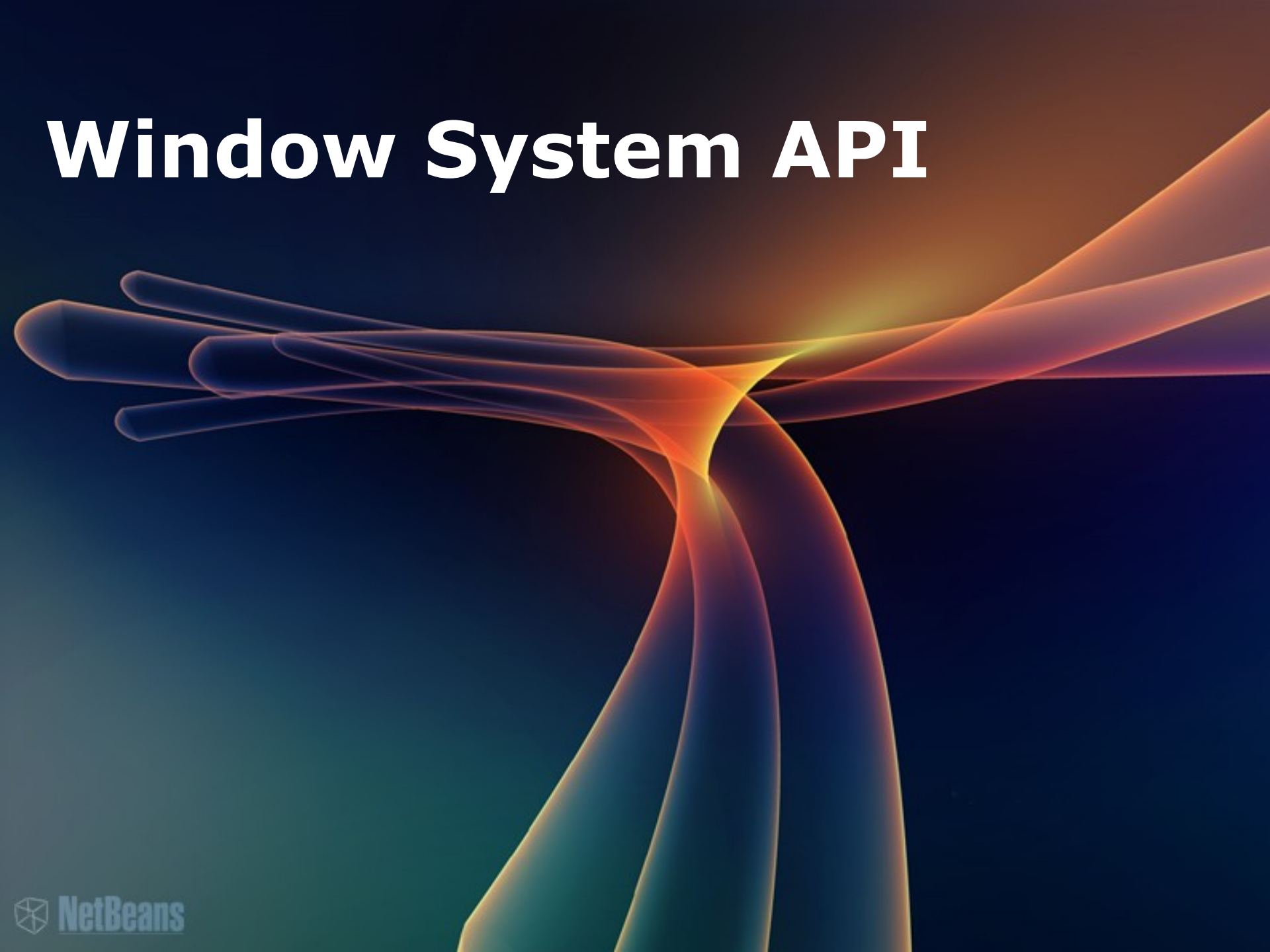
Window System API



Agenda

- 1. Terminology
- 2. TopComponent
- 3. WindowManager
- 4. Mode
- 5. TopComponentGroup
- 6. Configuration Options
- 7. Case Studies

1. Terminology

What is a window system?

A window system is a system for ***managing*** windows.

1. Terminology (cont.)

What does it look like?

Let's look at a few examples...

1. Terminology (cont.)

So why do I need it?

Otherwise your application will have one frame... or... you will have a LOT of work.

1. Terminology (cont.)

What alternatives are there?

JIDE, VLDocking, FlexDock, MyDoggy, JDocking, and Others.

1. Terminology (cont.)

What are the **3** main things I should know about the NetBeans window system?

(1) Only 4 API classes (1 subclassable)
(2) Many XML configuration files
(3) Mostly the defaults should be ok

1. TopComponent

- TopComponent = JPanel
- Provides window for application
- 6 themes for discussion
  - Logical window management
  - Creating a TopComponent
  - TopComponent lifecycle
  - State
  - Persistence
  - Limiting the window system's behavior
- Limitations

1. TopComponent (cont.)

- Logical window management
  - Moving/repositioning
  - Docking/undocking
  - Snapping
  - Transparency
  - Context sensitivity
  - Group behavior
  - Special effects (experimental)

1. TopComponent (cont.)

- Creating a TopComponent.
  - How to create a TopComponent
  - Looking at the generated files
  - Using Matisse GUI Builder for design

1. TopComponent (cont.)

- TopComponent lifecycle methods:
  - requestVisible()
  - requestActive()
  - componentHidden()
  - componentShowing()
  - componentDeactivated()
  - componentActivated()
  - componentClosed()
  - componentOpened()

1. TopComponent (cont.)

- State
  - opened
  - closed
  - minimized
  - maximized
  - docked
  - undocked
- Full screen mode (Alt-Shift-Enter)
- Layout reset

1. TopComponent (cont.)

- Persistence
  - Upon restart, last state of TopComponent persists
  - java.io.Externalizable is used
  - Skeleton code generated by wizard
  - You can modify the default persistence code

1. TopComponent (cont.)

- Persistence Modes
  - PERSISTENCE_ALWAYS
  - PERSISTENCE_NEVER
  - PERSISTENCE_ONLY_OPENED

1. TopComponent (cont.)

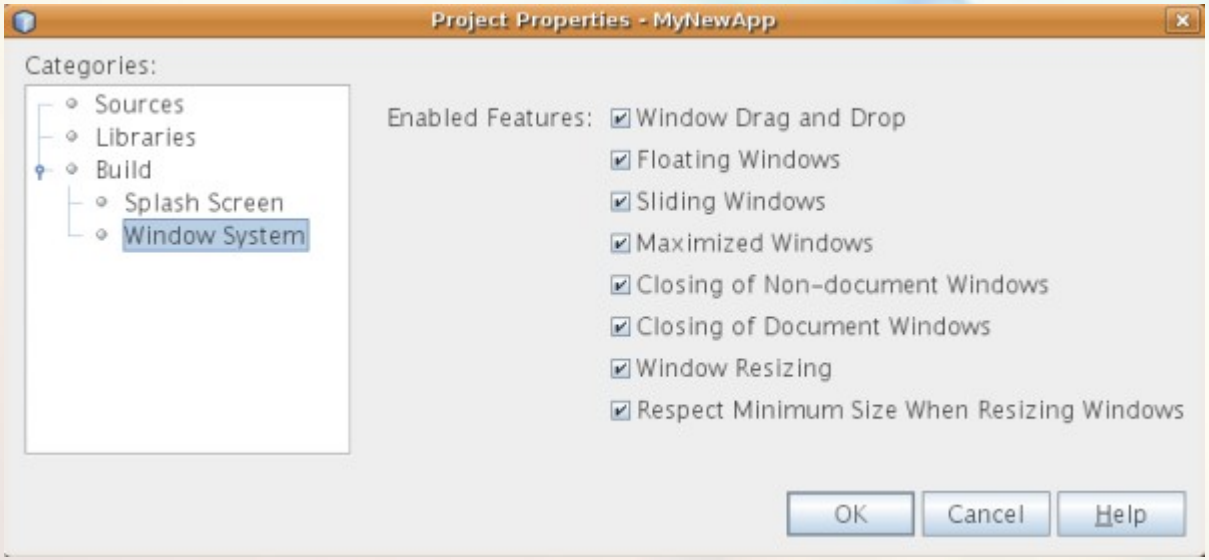
```
@Override
public Object writeReplace() {
    return new ResolvableHelper();
}

final static class ResolvableHelper implements Serializable {

    private static final long serialVersionUID = 1L;

    public Object readResolve() {
        return DemoTopComponent.getDefault();
    }

}
```

1. TopComponent (cont.)

```
@Override
public Object writeReplace() {
    return new ResolvableHelper(nameField.getText());
}

final static class ResolvableHelper implements Serializable {

    private static final long serialVersionUID = 1L;
    private final String name;

    private ResolvableHelper(String text) {
        this.name = text;
    }

    public Object readResolve() {
        DemoTopComponent result = DemoTopComponent.getDefault();
        result.nameField.setText(name);
        return result;
    }
}
```

1. TopComponent (cont.)

- Limiting the window system's behavior



1. TopComponent (cont.)

- Limitations
  - Easy to make changes for whole window system; tricky to make changes for specific window.

1. TopComponent (cont.)

- Limitations
  - Easy to make changes for whole window system; tricky to make changes for specific window.
  - Not easy to replace NetBeans window system with your own window system (unlike Spring RCP).

1. TopComponent (cont.)

- Limitations
  - Easy to make changes for whole window system; tricky to make changes for specific window.
  - Not easy to replace NetBeans window system with your own window system (unlike Spring RCP).
  - Not all events can currently be caught, e.g., pin event.

Agenda

- 1. Terminology
- 2. TopComponent
- 3. WindowManager
- 4. Mode
- 5. TopComponentGroup
- 6. Configuration Options
- 7. Case Studies

3. Window Manager

- Overall manager for state of user interface
  - Rare to write code that touches this class
  - Simply fetch a reference to the main window
  - For example, ask the registry for the registered TopComponents

3. Window Manager

- findTopComponent()
- findTopComponentGroup()
- getMainWindow()
- getRegistry()

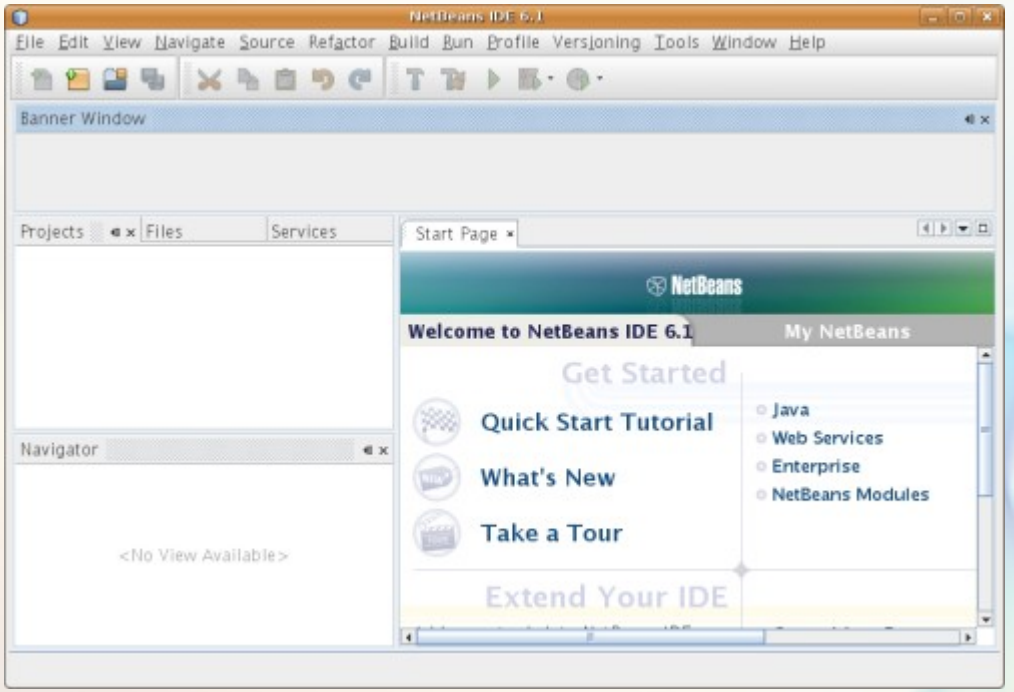
3. Window Manager

```
OutputWriter writer;
InputOutput io = IOProvider.getDefault().getIO("Opened", false);
writer = io.getOut();
io.select();

Set<TopComponent> tcs =
    WindowManager.getDefault().getRegistry().getOpened();
 for (TopComponent topComponent : tcs) {
     writer.println(topComponent.getName() + "\n");
 }
```

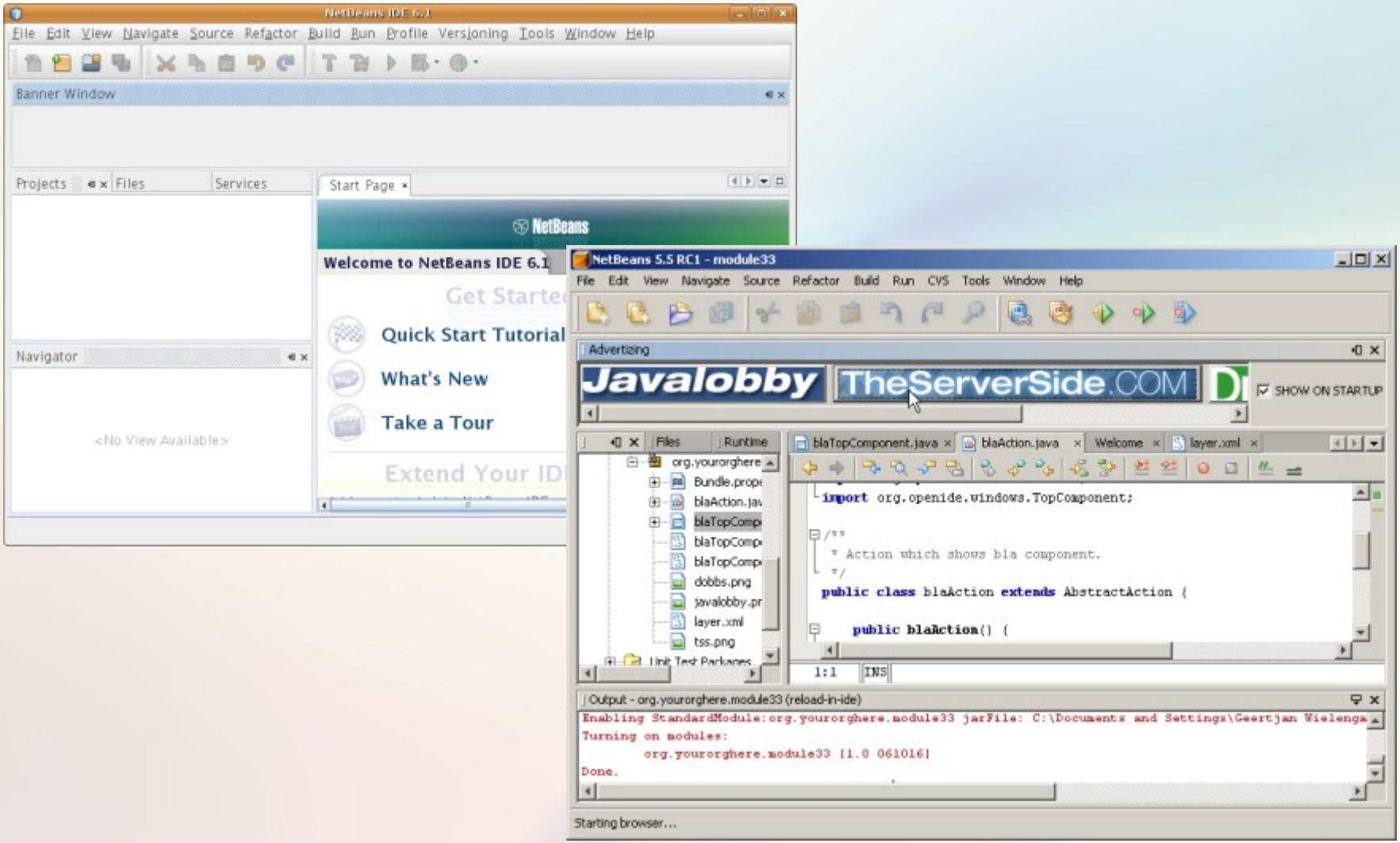
4. Mode

- Mode = position in application
  - Many defaults
  - Not common to create your own...
  - Can create custom modes declaratively
  - Window System Mode file
  - Generate XML and copy their content
  - Can dock into them programmatically

4. Mode


NetBeans IDE 6.1
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Banner Window
Projects
Files
Services
Navigator
<No View Available>
Start Page
NetBeans
Welcome to NetBeans IDE 6.1
My NetBeans
Get Started
Quick Start Tutorial
What's New
Take a Tour
Java
Web Services
Enterprise
NetBeans Modules
Extend Your IDE

4. Mode


NetBeans IDE 6.1
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Banner Window
Projects Files Services
Navigator
<No View Available>
Start Page
NetBeans
Welcome to NetBeans IDE 6.1
Quick Start Tutorial
What's New
Take a Tour
NetBeans 5.5 RC1 - module33
File Edit View Navigate Source Refactor Build Run CVS Tools Window Help
Advertising
Javalobby
TheServerSide.COM
SHOW ON STARTUP
Files Runtime
org.yourorghere
Bundle.prop
blaAction.jav
blaTopComp
dobbs.png
javalobby.pr
layer.xml
tss.png
blaTopComponent.java
blaAction.java
Welcome
layer.xml
import org.openide.windows.TopComponent;
/**
* Action which shows bla component.
*/
public class blaAction extends AbstractAction {
public blaAction() {
1:1 INS
Output - org.yourorghere.module33 (reload-in-ide)
Enabling StandardModule:org.yourorghere.module33 jarFile: C:\Documents and Settings\Geertjan Wielenga
Turning on modules:
org.yourorghere.module33 [1.0 061016]
Done.
Starting browser...

4. Mode

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mode PUBLIC
        "-//NetBeans//DTD Mode Properties 2.1//EN"
        "http://www.netbeans.org/dtds/mode-properties2_1.dtd">
<mode version="2.3">
   <name unique="bannerMode" />
   <kind type="view" />
   <state type="joined" />
   <constraints>
      <path orientation="horizontal" number="20" weight="0.3"/>
      <path orientation="vertical" number="21" weight="0.25"/>
   </constraints>
   <bounds x="0" y="0" width="0" height="0" />
   <frame state="0"/>
   <active-tc  id="BannerTopComponent" />
   <empty-behavior permanent="false"/>
</mode>
```

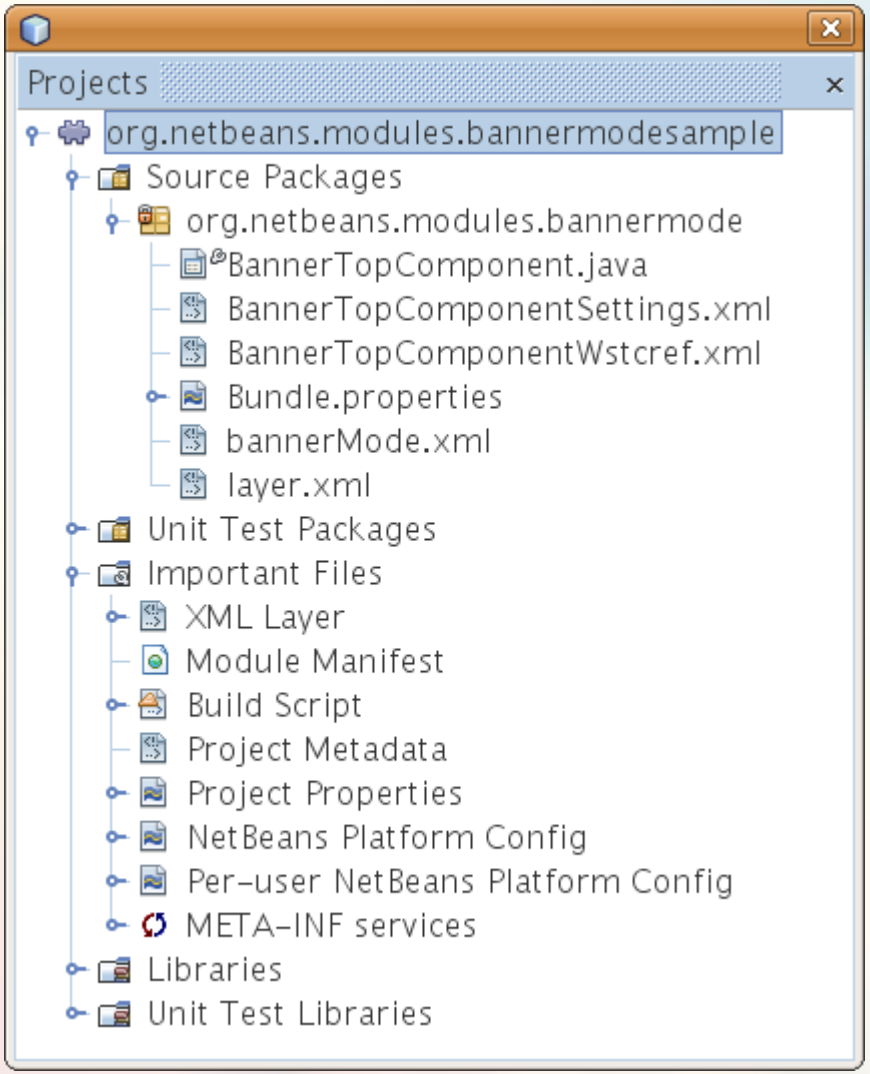
4. Mode

```
<folder name="Windows2">
    <folder name="Components">
        <file name="BannerTopComponent.settings"
                url="BannerTopComponentSettings.xml"/>
    </folder>
    <folder name="Modes">
        <file name="bannerMode.wsmode" url="bannerMode.xml"/>
        <folder name="bannerMode">
            <file name="BannerTopComponent.wstcref"
                    url="BannerTopComponentWstcref.xml"/>
        </folder>
    </folder>
</folder>
```

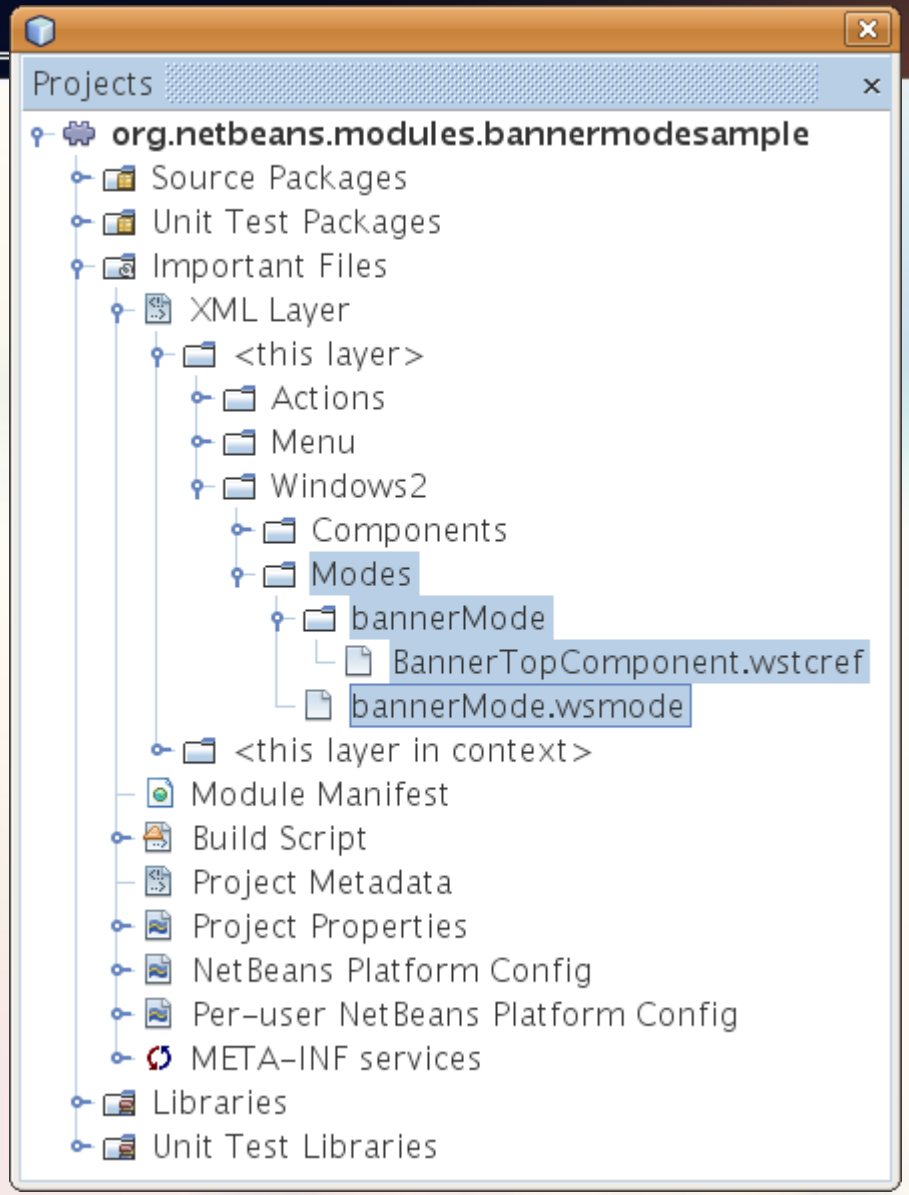
4. Mode

```
@Override
public void open() {

    Mode mode = WindowManager.getDefault().findMode(“bannerMode”);
    if (mode != null) {
        mode.dockInto(this);
    }
    super.open();

}
```

4. Mode


Projects
org.netbeans.modules.bannermodesample
Source Packages
org.netbeans.modules.bannermode
BannerTopComponent.java
BannerTopComponentSettings.xml
BannerTopComponentWstcref.xml
Bundle.properties
bannerMode.xml
layer.xml
Unit Test Packages
Important Files
XML Layer
Module Manifest
Build Script
Project Metadata
Project Properties
NetBeans Platform Config
Per-user NetBeans Platform Config
META-INF services
Libraries
Unit Test Libraries

4. Mode


Projects
org.netbeans.modules.bannermodesample
Source Packages
Unit Test Packages
Important Files
XML Layer
<this layer>
Actions
Menu
Windows2
Components
Modes
bannerMode
BannerTopComponent.wstcref
bannerMode.wsmode
<this layer in context>
Module Manifest
Build Script
Project Metadata
Project Properties
NetBeans Platform Config
Per-user NetBeans Platform Config
META-INF services
Libraries
Unit Test Libraries

5. TopComponentGroup

- Opening of 1 TopComponent triggers opening of other related TopComponents
  - Window System Group
  - Window System TopComponent Group

5. TopComponentGroup

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE group PUBLIC "-//NetBeans//DTD Group Properties 2.0//EN"
"http://www.netbeans.org/dtds/group-properties2_0.dtd">

<group version="2.0">
    <module name="org.netbeans.modules.windowgroupsample" spec="1.0" />
    <name unique="MyGroup" />
    <state opened="false" />
</group>
```

5. TopComponentGroup

```
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE tc-group PUBLIC "-//NetBeans//DTD Top Component in Group Properties
    2.0//EN"
"http://www.netbeans.org/dtds/tc-group2_0.dtd">

<tc-group version="2.0">
    <module name="org.netbeans.modules.windowgroupsample" spec="1.0"/>
    <tc-id id="OneTopComponent" />
    <open-close-behavior open="true" close="true" />
</tc-group>
```

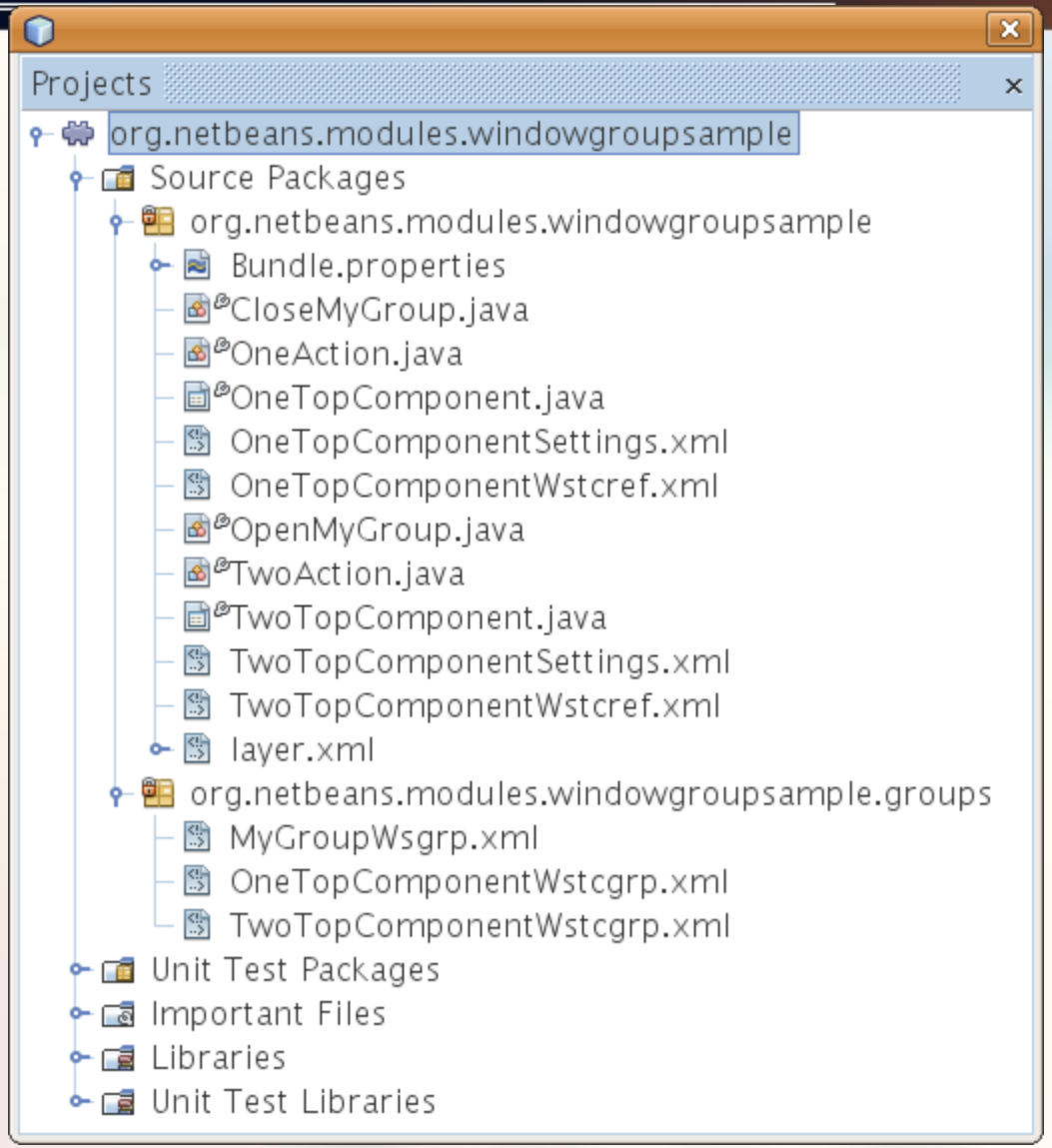
5. TopComponentGroup

```
<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE tc-group PUBLIC "-//NetBeans//DTD Top Component in Group Properties
    2.0//EN"
"http://www.netbeans.org/dtds/tc-group2_0.dtd">

<tc-group version="2.0">
    <module name="org.netbeans.modules.windowgroupsample" spec="1.0"/>
    <tc-id id="TwoTopComponent" />
    <open-close-behavior open="true" close="true" />
</tc-group>
```

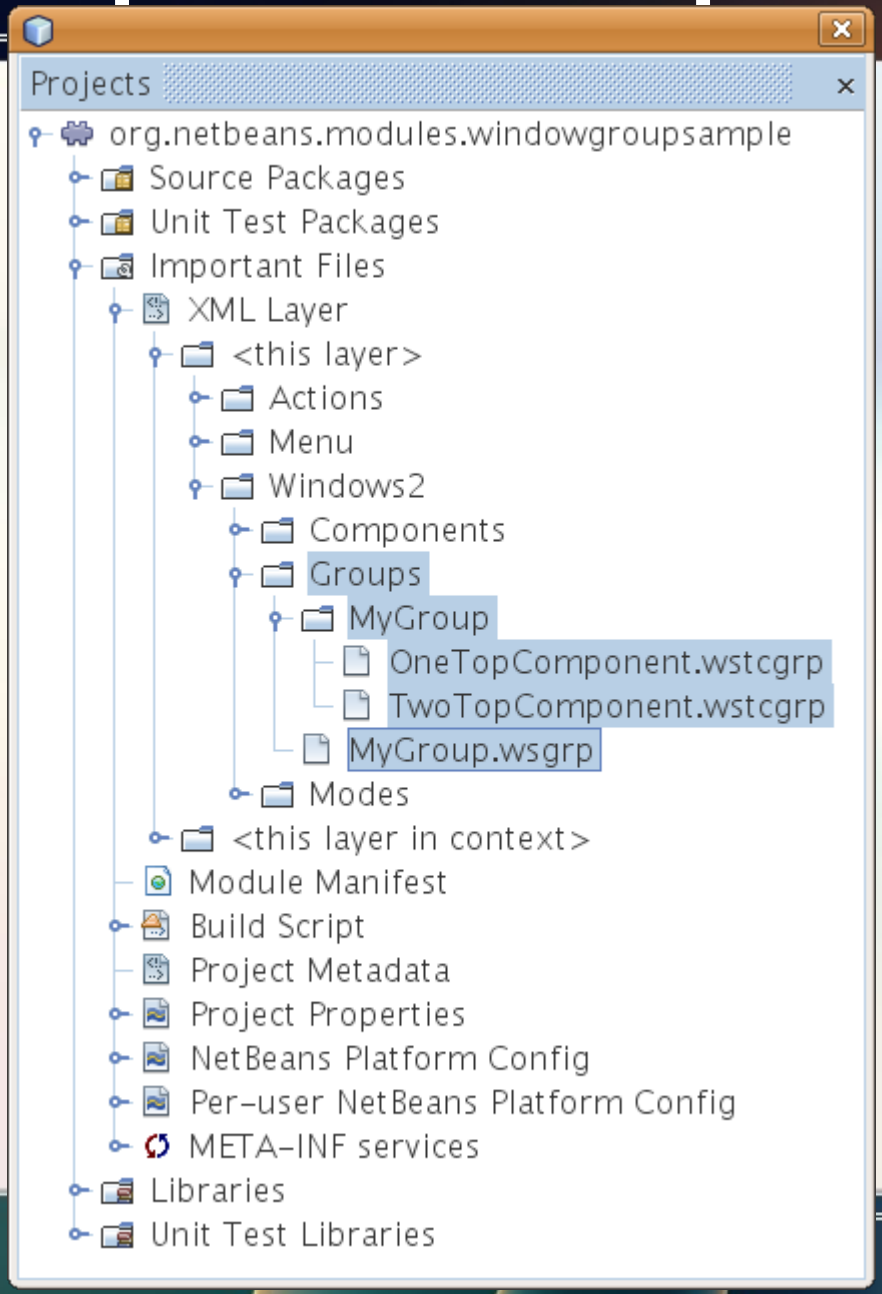
5. TopComponentGroup

```
<folder name="Windows2">
    <folder name="Components">
        <file name="OneTopComponent.settings" url="OneTopComponentSettings.xml"/>
        <file name="TwoTopComponent.settings" url="TwoTopComponentSettings.xml"/>
    </folder>
    <folder name="Modes">
        <folder name="editor">
            <file name="TwoTopComponent.wstcref" url="TwoTopComponentWstcref.xml"/>
        </folder>
        <folder name="output">
            <file name="OneTopComponent.wstcref" url="OneTopComponentWstcref.xml"/>
        </folder>
    </folder>
    <folder name="Groups">
        <file name="MyGroup.wsgrp" url="groups/MyGroupWsgrp.xml"/>
        <folder name="MyGroup">
            <file name="OneTopComponent.wstcgrp" url="groups/OneTopComponentWstcgrp.xml"/>
            <file name="TwoTopComponent.wstcgrp" url="groups/TwoTopComponentWstcgrp.xml"/>
        </folder>
    </folder>
</folder>
```

5. TopComponentGroup



5. TopComponentGroup



6. Configuration Options

- http://bits.netbeans.org/dev/javadoc/org-openide-windows/architecture-summary.html
  - netbeans.winsys.tc.keep_preferred_size_when_slided_in
  - TopComponentAllowDockAnywhere
  - netbeans.winsys.imageSource
  - -J-Dnetbeans.winsys.hideEmptyDocArea=true
  - ....

7. Tips & Tricks

1. Look in the layer explorer in the IDE
2. Read the DTD descriptions
3. NetBeans Zone
4. http://blogs.sun.com/geertjan

8. Case Studies

1. Create a welcome screen
   - Undocked at startup
   - Modal
   - Remove actions
   - Change close behavior to dock into right sliding side.

8. Case Studies

1. Create a welcome screen
   - Undocked at startup
   - Modal
   - Remove actions
   - Change close behavior to dock into right sliding side.
2. Create two TopComponents that open and close together.

Questions & Answers