

NetBeans ~ JavaBeans



Jaroslav Tulach

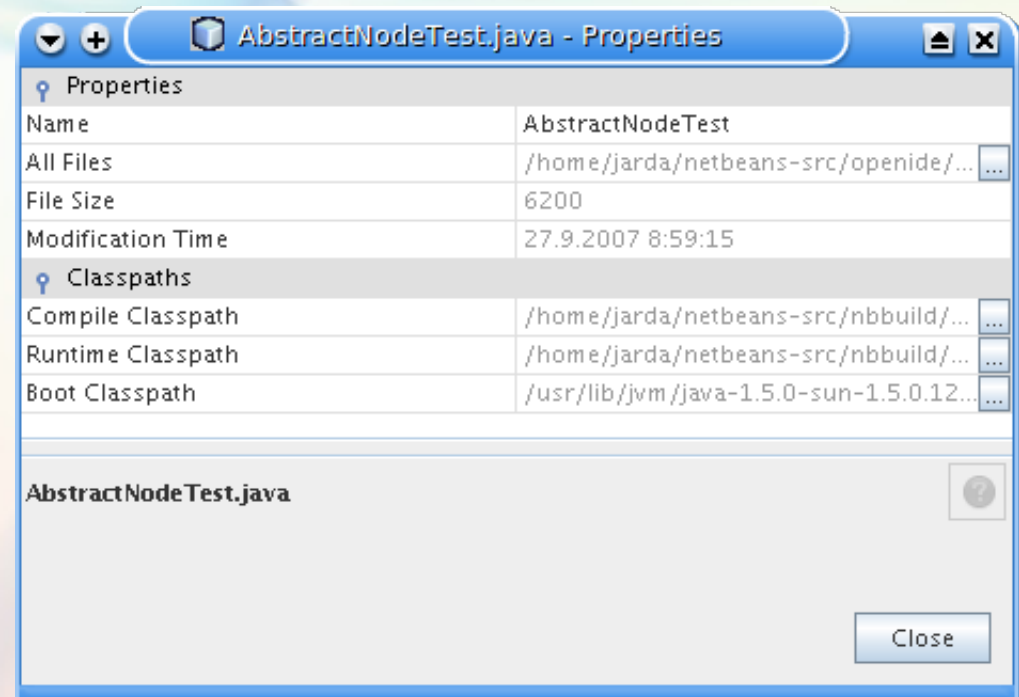
Sun Microsystems

Agenda

- Why NetBeans is called NetBeans?
- Nodes
- Node views
- Composition
- Q/A

JavaBeans

- JavaBeans for network
- Beans everywhere 1.0
 - > bean context
 - > property sheet
- Problems
 - > API vs. SPI



Nodes

- Types JavaBeans
 - > no reflection
 - > standard listeners
 - > extensibility
- Support for hierarchy
 - > correctness guaranteed
- Bridge to beans via BeanNode

Nodes API

```
import org.openide.nodes.AbstractNode;  
import org.openide.nodes.Children;  
class MyNode extends AbstractNode {  
    public MyNode() {  
        super(new MyChildren());  
    }  
}  
class MyChildren extends Children.Keys<String> {  
    protected void addNotify() {  
        setKeys(Collections.nCopies(1, "Child"));  
    }  
    protected Node[] createNodes(String key) {  
        MyNode n = new MyNode();  
        n.setName(key);  
        return new Node[] { n };  
    }  
}
```

Rules

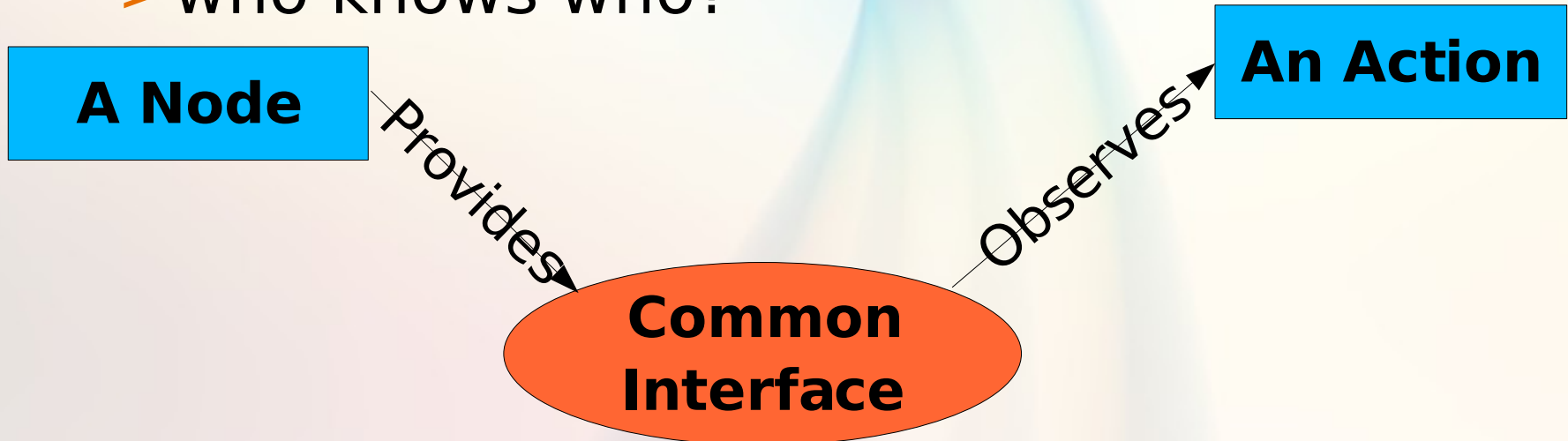
- Nodes are the model
 - > not the data
- Create them lazily
 - > Children.addNotify, Children.setKeys
- Make sure they garbage collect
 - > leaks with listeners
 - > possible use of removeNotify

Node Properties

- extends `java.beans.FeatureDescriptor`
- Clipboard operations, D'n'D
- Actions, Customizer
- Help, Icon, HTML name
- Persistence
- Properties

Nodes Actions

- Addition over JavaBeans
- Swing Actions
 - > `Action[] Node.getAction(boolean)`
- Multiselection
 - > who knows who?



Nodes Context

- Lookup `Node.getLookup()`
 - > Passed in constructor
- Replacement for old `getCookie(Class)`
 - > No marker interface
- `OpenCookie`, `EditorCookie`, etc.
- Multiselection
 - > `ProxyLookup`

Context Actions

- <http://wiki.netbeans.org/wiki/view/DevFaqActionCodeContextSensitive>

```
public class FooAction extends AbstractAction implements LookupListener, ContextAwareAction {
    private Lookup context;
    Lookup.Result lkpInfo;
    public FooAction() {
        this(Utilities.actionsGlobalContext());
    }
    private FooAction(Lookup context) {
        this.context = context;
    }
    void init() {
        Lookup.Template tpl = new Lookup.Template(Whatever.class);
        lkpInfo = context.lookup (tpl); lkpInfo.addLookupListener(this); resultChanged(null);
    }
    public boolean isEnabled() {
        init(); return super.isEnabled();
    }
    public Action createContextAwareInstance(Lookup context) {
        return new FooAction(context);
    }
}
```

Root Container

```
class MyPanel extends JPanel implement
    ExplorerManager.Provider {

public MyPanel() {
    myManager = new ExplorerManager();
    add(new BeanTreeView());
    add(new PropertySheetView());
    myManager.setRootContext(myNode);
}

public ExplorerManager getExplorerManager() {
    return myManager;
}
```

Views

- ExplorerManager
 - > root context
 - > explored context
 - > selected nodes (vetoable)
- General Model behind Swing
 - > BeanTreeView, ContextTreeView
 - > ListView
 - > PropertySheet
 - > TableTreeView

Write your own View

- Just a visual JavaBean
- Overwrite addNotify and removeNotify
 - > search parents for ExplorerManager.Provider
 - > add listeners
 - > display what ExplorerManager says
- Control ExplorerManager
 - > call setters
 - > add vetoable listeners

Conclusion

- Nodes are typed JavaBeans
- Hierarchy
- Extensible
- Rich Set of Views
- Standalone

DEMO

- Standalone Explorer

Q&A

- <http://bits.netbeans.org/dev/javadoc/org-openide-nodes/>
- <http://bits.netbeans.org/dev/javadoc/org-openide-explorer/>