

Data Objects & Editors



David Strupl
Staff Engineer
Sun Microsystems

Agenda

- DataObject
- Factories
- Presentation
- Editor APIs
 - > Schliemann
 - > Lexer
 - > Folding
 - > Highlighting
 - > Code completion
 - > ...

DataObject

- Logical object around a FileObject
 - > Understands the file's content
 - > Copy, move, shadow, createFromTemplate
 - > Can represent multiple files (not recommended)
- Factories
 - > A module installs a *DataLoader* for a particular file type
 - > Usually registered by MIME type + file extension
 - > That loader is a factory for DataObjects for that type of file

DataObjects provide

- > A Lookup – you can query for objects you know about
- > A Node – to represent it in the UI
- > A way to have an object that survives file rename/move

DataLoader

- Factory
 - > by mime type
 - > by extension
- Context Actions
 - > String actionsContext();
 - > Path to a folder in the system filesystem (layer.xml file)
 - > Can be regular javax.swing.Actions
- Can have persistent state

FileSystems vs. Loaders API

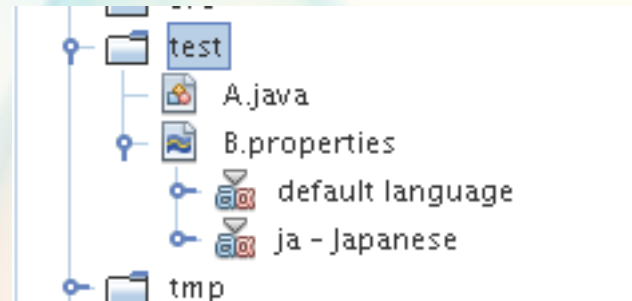
- One type vs. multiple types
- One file vs. group of files
- Raw data vs. operations on content
 - > Think of a DataObject as representing a (potentially) *parsed* file

A.java

A.class

B.properties

B_ja.properties



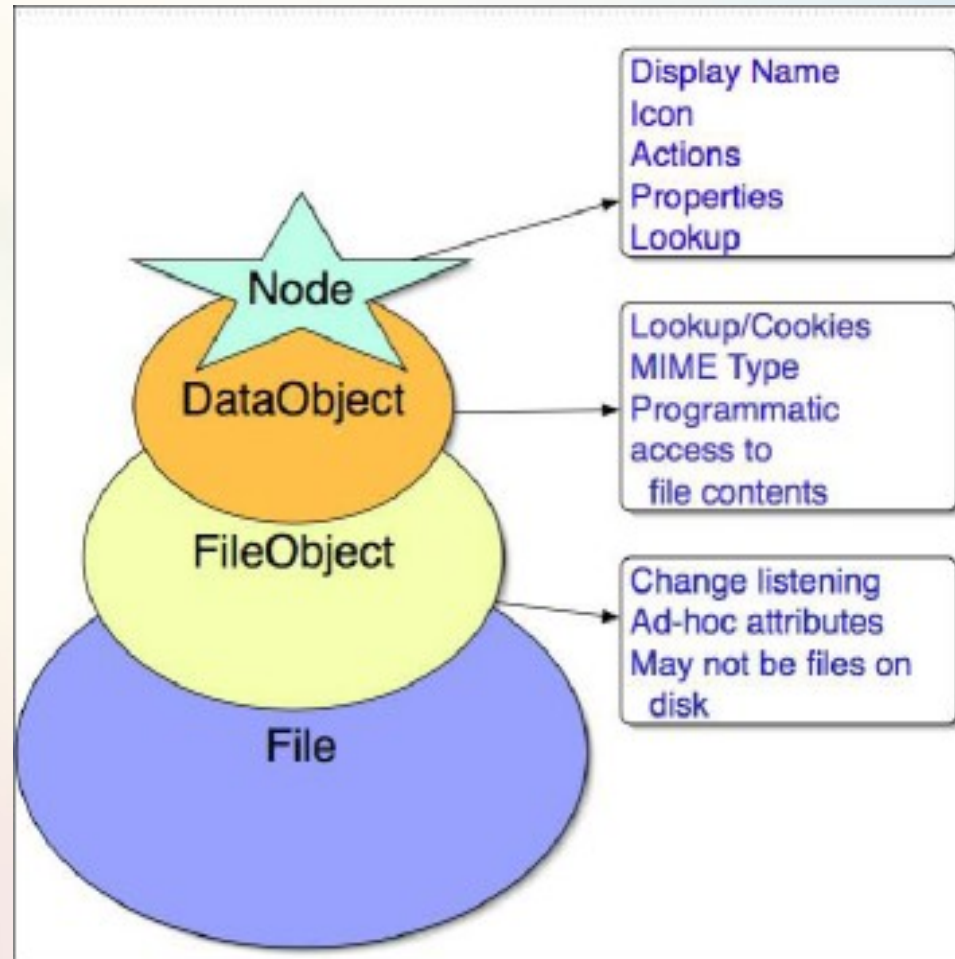
Visualizing DataObjects

- A DataObject has a...

NODE!

```
DataObject.getNodeDelegate()
```

Diagram of API Layers



Files, DataObjects, Nodes, Explorer

- One set of APIs – but two usages:
 - > 1. Visualizing and interacting with real files on disk
 - > 2. Visualizing and interacting with parts of NetBeans configuration

Files on disk are a filesystem.

The “system filesystem” is also a filesystem

A registry of configuration information, read/write

Many pieces of UI in NetBeans are just a view of some folder in the System Filesystem

Menus, Toolbars, nodes in the Services tab

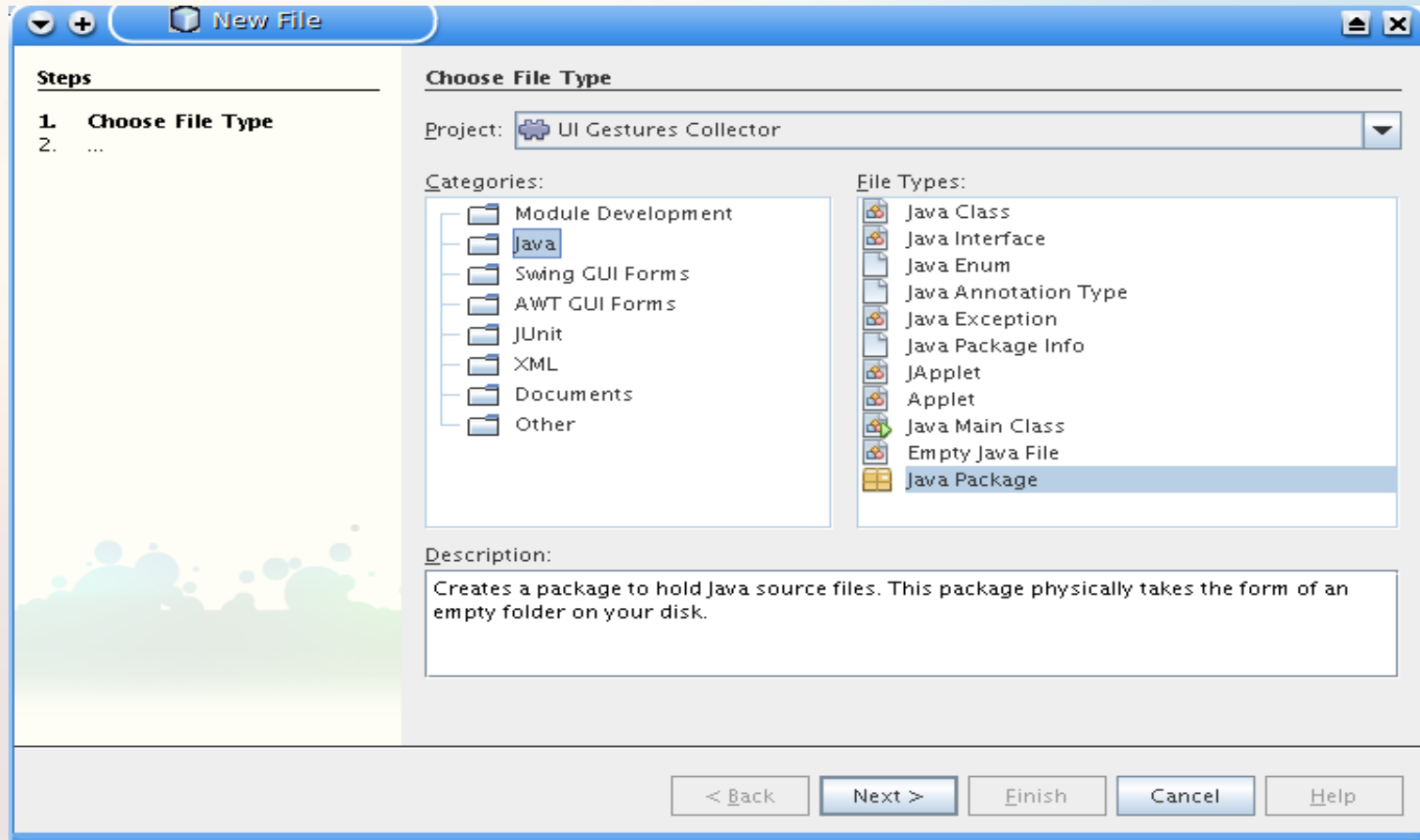
DataObject Classes

- DataObject
 - > base API class, default data object
- MultiDataObject
 - > abstract support class - may represent >1 file
- DataFolder
- DataShadow - like a unix symbolic link
- InstanceDataObject - represents an object
 - > settings management - .instance files

register an object in the system
filesystem, different folders have

Visual Creation

- /Templates && TemplateWizard



Templating Languages

- Any scripting language
 - > javax.script.ScriptEngine
 - > <http://scripting.dev.java.net>
- Support for Freemarker
 - > substitutions
 - > variables
 - > cycles
- CreateFromTemplateAttributeProvider
- TemplateWizard.putProperty(xxx) -> \$
{wizard.xxx}

Visual Representation

- Represented by Nodes
 - > DataObjects are the model
 - > Nodes are the *presentation* of the model
 - > They add human-friendly attributes - name, icon, etc.
- `DataObject.getNodeDelegate()`
 - > `DataNode` - premade superclass
 - > `FolderNode` & `FolderChildren`
- Write you own hierarchy
 - > projects
 - > files
 - > favorites

Context of a DataObject

- `DataObject.getLookup()`
 - > added in 6.0 – previously
 - > `DataObject.getCookie(Class type)` or `DataObject.getNodeDelegate().getLookup()`
- In case of `MultiDataObject`
 - > `getCookieSet().getLookup()`
- Can be controlled by
 - > `getCookieSet().assign(Data.class, data)`

Text Editor

- API for accessing
 - > EditorCookie
 - > OpenCookie
 - > EditCookie
- Supporting classes
 - > CloneableEditorSupport
 - > CloneableEditor
 - > DataEditorSupport

Editor APIs

- Syntax coloring
- Code folding
- Navigation
- Code completion
- Brace matching
- Actions
- Tooltips
- Hyperlinks
- Indentation

• Annotations (e.g. error lines)

- NBS files
- Simple languages
 - > LL(k)
- “Real” languages support by the IDE use editor APIs directly

Conclusion

- DataObject
 - > logical view over a file
 - > created by factories
- Extensible via layers
 - > actions
 - > templates
- Visualized by Nodes
- Content edited by editors

DEMO

- File Type Integration