

NetBeans Platform

Tom Wheeler

11/11/09

SIUE :: November 2009

Intro: What I Cover in this Section

- What the DataSystems API
- How it relates to the FileSystems API
- What DataObjects and DataLoaders are
- How to add support for a new file type
- What instance and shadow files are
- Handy utility methods for conversion

DataSystems API

- FileSystems API is about bytes
 - ◇ Not information, just raw bytes
- DataSystems is a higher level API
 - ◇ For getting information from FileObjects
 - ◇ It depends on FileSystems API

DataObjects

- DataObject: The most important class
 - ◇ It's a “recognized” file FileObject
 - ◇ This means platform knows how to process it
 - ◇ By parsing its content
 - ◇ And adding object representing content in Lookup
 - ◇ Typically constructed from just one FileObject
 - ◇ Such as a JPEG image or an XML document
 - ◇ But can actually be composed of multiple FOs
 - ◇ Such as .java/.class file or .java/.form file pair

DataLoaders

- Where do DataObjects come from?
 - ◇ DataLoaders are registered into platform
 - ◇ Recognize files based on MIME type
 - ◇ MIME type association typically by file ext.
 - ◇ But sometimes by inspecting file content (XML)
 - ◇ First DataLoader to recognize FileObj. wins

Other Important API Features

- DataSystems API also provides support for
 - ◇ DataNode (node representing a DataObject)
 - ◇ Templates
 - ◇ DataShadow (like a UNIX symlink)
 - ◇ Instance files
 - ◇ FileSystem operations
 - ◇ Copying/moving/deleting/renaming files
 - ◇ OpenSupport
 - ◇ Adds support for opening a file

Demo: Register Support for File Type

- I will demonstrate how
 - ◇ To add support for a new file type
 - ◇ With support for creating via template
 - ◇ This changes to layer file

Instance Files

- NB has a lot of performance optimizations
 - ◇ Need to show toolbars/menus quickly
 - ◇ Object construction can be slow
- Instance files are one such optimization
 - ◇ They represent an instance of an object
 - ◇ Allows you to inspect certain aspects
 - ◇ Without actually instantiating it
 - ◇ It's a form of lazy loading
 - ◇ They have a .instance extension

System FileSystem and Instance Files

- The SysFS is full of instance files
 - ◇ The Actions folder is a good example
 - ◇ Allows one to define an instance once
 - ◇ You can refer to this instance elsewhere
- How do you refer to the instance?
- Shadow files
 - ◇ Links a .instance file to another place in FS
 - ◇ Menu/Toolbar folders in SysFS use this a lot
 - ◇ Example on next slide

Instance and Shadow File Example

- Here I define a new action
 - ◇ And add it to the File menu

```
<folder name="Actions">
  <folder name="Edit">
    <file name="com-tomwheeler-example-SomeAction.instance"/>
  </folder>
</folder>

<folder name="Menu">
  <folder name="File">
    <file name="com-tomwheeler-example-SomeAction.shadow">
      <attr name="originalFile"
        stringvalue="Actions/Edit/com-tomwheeler-example-SomeAction.instance"/>
      <attr name="position" intvalue="100"/>
    </file>
  </folder>
</folder>
```

Handy Utility Methods

- Here's how to convert/access
 - ◇ File -> FileObject -> DataObject
 - ◇ DataObject -> FileObject
 - ◇ DataObject -> Node

```
// convert a disk-based file to a FileObject
File file = new File("/home/twheeler/example.txt");
FileObject fileObj = FileUtil.toFileObject(file);

// find the DataObject for this FileObject
DataObject dataObj = DataObject.find(fileObj);

// get the FileObject for a DataObject
FileObject fo = dataObj.getPrimaryFile();

// Get the node for the DataObject
Node node = dataObj.getNodeDelegate();
```

Review Questions

- How does DataSystems relate to FS API?
- How does a FileObject differ from a DataObject?
- Why are instance files used?
- What is a shadow file?

Recap

- DataSystems is a higher level API
 - ◇ For getting information, not just bytes
- DataObjects provide structured content
 - ◇ Usually from the Lookup
 - ◇ DataObjects are created by DataLoaders
 - ◇ They can represent one or more FileObjects
- Instance files are a performance trick
 - ◇ Shadow files allow multiple refs to same obj

Exercise (30 minutes)

- Create a new platform app based on IDE
 - ◇ Add support for a new file type
 - ◇ You can create a custom icon with Paint
 - ◇ When you run it
 - ◇ Open favorites window
 - ◇ Create a new file from template
 - ◇ See that it is recognized by the app!