Configuring UCLA Infrared Lab Java GUIs

Jason Weiss
weiss@astro.ucla.edu
2007-06-06

Java GUIs written by the UCLA Infrared Lab (for example, OSIRIS GUIs, OSIRIS Data Reduction File GUI, MOSFIRE GUIs) are written such that select settings of the program can be specified and modified in an Extensible Markup Language (XML) configuration file. This design allows the GUI to be customized in certain ways without the need for recompilation. Settings can quantify many things, such as the default size and location of the GUI on the screen, the type of font and color used for certain labels, and the default directories to open or save files.

Configuration files are specified as an argument to the GUI during runtime using the –C tag:

```
> java App –C config.xml
```

The Parameters and XmlToParams Classes

All of the constants used by a program are stored in a class typically called `<App>Parameters`, where `<App>` is the name of the program, defined in a file called `<App>Parameters.java`. For example, the ODRFGUI has a class called `ODRFGUIParameters` defined in a class called `ODRFGUIParameters.java`. This class defines each constant with a name, datatype, and usually a default value. The GUI uses the values of the members in this class during execution.

Unless otherwise specified, knowledge of the Parameters class and the members within is required to determine which settings are available to override.

The `XmlToParams` class is the class with the intelligence to extract the settings defined in the XML configuration file and set them in a Parameters object.

XML Configuration Files

All parameters (of the supported datatypes, see below) in the Parameters object can be overridden by specifying a new value of the parameter in the XML configuration file. Each XML tag must have the `“datatype”` as the name of the tag, and have the attribute `"paramName"` which identifies the field in the Parameters object to receive the value of the parameter.

Regardless of the various datatypes in the parameters file, the only configurable datatypes are: primatives, except char (`boolean`, `int`, `long`, `float`, `double`), `String`, `java.awt.Color`, `java.awt.Dimension`, `java.io.File`, `java.awt.Font`, and

`java.awt.Point. String[]` arrays are also allowed, using `StringArray` as the name of the tag.

XML tags must be defined as follows:

- All tags must have a `paramName` attribute, with the value mapped to the name of the corresponding Parameter object field name. The parameter name is case-sensitive.
- `Color` tags must have `"red"`, `"green"`, and `"blue"` attributes, each set to integer values between 0 and 255.
- `Dimension` tags must have `"xs"` and `"ys"` attributes, each set to positive integer values.
- `File` tags must have a `"value"` attribute, set to the path of the file or directory. Value attributes that start with a tidle (“~”) replace the tilde with the current user’s home directory.
- `Font` tags must have a `"name"` attribute, set to the name of the font; a `"style"` attribute, describing the face of the font, set to one of "plain", "bold", "italics", or "bold+italics"; and a `"size"` attribute, set to an integer value dictating the size in points of the font.
- `Point` tags must have `"x"` and `"y"` attributes, each set to positive integer values.
- `String` and primitive tags must have a single `"value"` attribute giving the value of the paramater.
- `StringArray` tag must have children tags of type `String`. The `paramName` tag is an attribute of the StringArray tag. The children `String` tags do not specify a `paramName` but must have a `"value"` attribute.

Example

In this example, values in the configuration file are set to specify the title and size of a dialog, and the contents of a list widget, showing a list of fruits.

- in XML File ("example.xml"):

```
...
<String paramName="TITLE_DIALOG" value="XML Example Dialog" />
<Dimension paramName="DIM_DIALOG" xs="400" ys="300" />
<StringArray paramName=”FRUIT_LIST”>
 <String value=”Apple” />
 <String value=”Orange” />
 <String value=”Pear” />
</StringArray>
...
```

- in Java Parameter Class (AppParameters):

```
public class AppParameters {
  ...
  public static String TITLE_DIALOG = “App”;
```

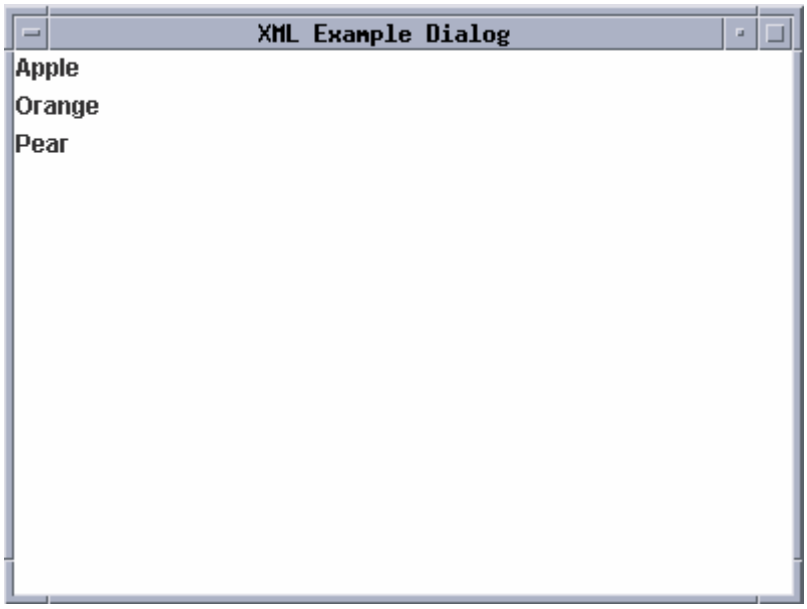
```
  public static java.awt.Dimension DIM_DIALOG = new
    java.awt.Dimension(200,100);
  public static String[] FRUIT_LIST;

  public ParamClass() {}
  ...
}
```

- in calling Java program:

```
...
AppParameters myParams = new AppParameters();
JDialog myDialog;
JList myList;

try {
  XmltoParams.extractParams(new File("example.xml"), myParams);

  myDialog = new JDialog(null, myParams.TITLE_DIALOG, false);
  myDialog.setSize(myParams.DIM_DIALOG);

  if (myParams.FRUIT_LIST != null)
    myList = new JList(myParams.FRUIT_LIST);
  else
    myList = new JList(new String[] {“No fruits today.”});

} catch (JDOMException e) {
...
```

The default dialog without the configuration applied would look like this:

With the configuration file, the dialog now looks like this:


XML Example Dialog
Apple
Orange
Pear