
BlueJ class templates



BlueJ already comes with a number of predefined class and method templates that you can use to create either a new class or a new method.

These templates are customizable and as such can be used as a starting point, when you need to add a new class or method template.

Here I will describe the steps for adding a new class template, but the steps for a method are similar.

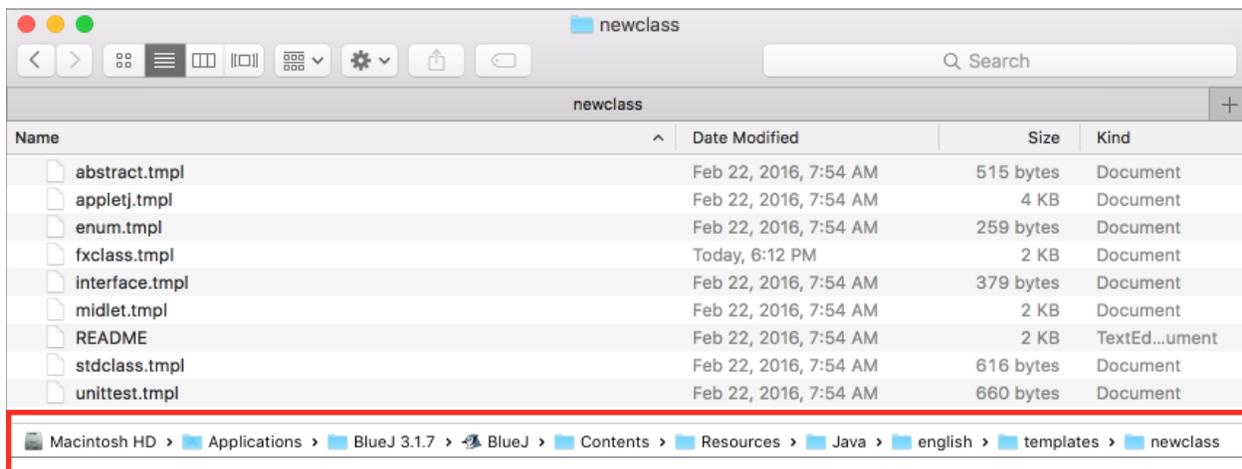
There are basically three steps to this process: 1) copy an existing template 2) modify it to meet your needs and 3) make the template available.

1) Copy an existing template.

The templates are in `<blueJ>/lib/<language>/templates/newclass`. Here `<blueJ>` is the installation directory on your file system, and `<language>` is the locale language you are using, e.g. english, french etc.

Navigate to this directory, find a template file (.tmpl) you want to copy and make a duplicate of it in the same directory. The screen capture below shows a MacOS folder after a copy of the `stdclass.tmpl` was made and named `fxclass.tmpl`.

Note the path for the MacOS.



2) Modify it to meet your needs.

You may now remove or add code/comments to this template to suit your specific class template needs. You may always return and make additional changes, so not fret if you don't get it right the first time.

```
fxclass.tmpl
$PKGLINE
/* Add the needed import statements to build your UI. */
import javafx.embed.swing.JFXPanel;
import javafx.application.Platform;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.control.Button;

/**
 * This is the JavaFX template code to be used as a starting point. BlueJ
 * does not fully support JavaFX applications, so this is the work around.
 *
 * Modify the buildUI() template to suit your needs.
 * Use the launchFX() and main() templates as is. No need to modify really.
 *
 * To run the application, simply execute the main() method on the class.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class $CLASSNAME {

    public void buildUI() {
        // NOTE: This method can be customized to your specific needs.

        // Create the stage.
        Stage primaryStage = new Stage();
        Scene scene = new Scene( new Button("Welcome to JavaFX"), 300, 300 );

        primaryStage.setTitle("Welcome to JavaFX");
        primaryStage.setScene(scene);
        primaryStage.show();
    } // end buildUI()

    public void launchFX() {
        // NOTE: This method should not be changed.

        // Initialize JavaFX.
        new JFXPanel();

        // Makes sure that JavaFX doesn't exit when the first window is closed.
        Platform.setImplicitExit(false);

        // Runs initialization on the JavaFX thread.
        Platform.runLater( () -> buildUI() );
    } // end launchFX()

    public static void main(String[] args) {
        $CLASSNAME app = new $CLASSNAME();
        app.launchFX();
    } // end main()

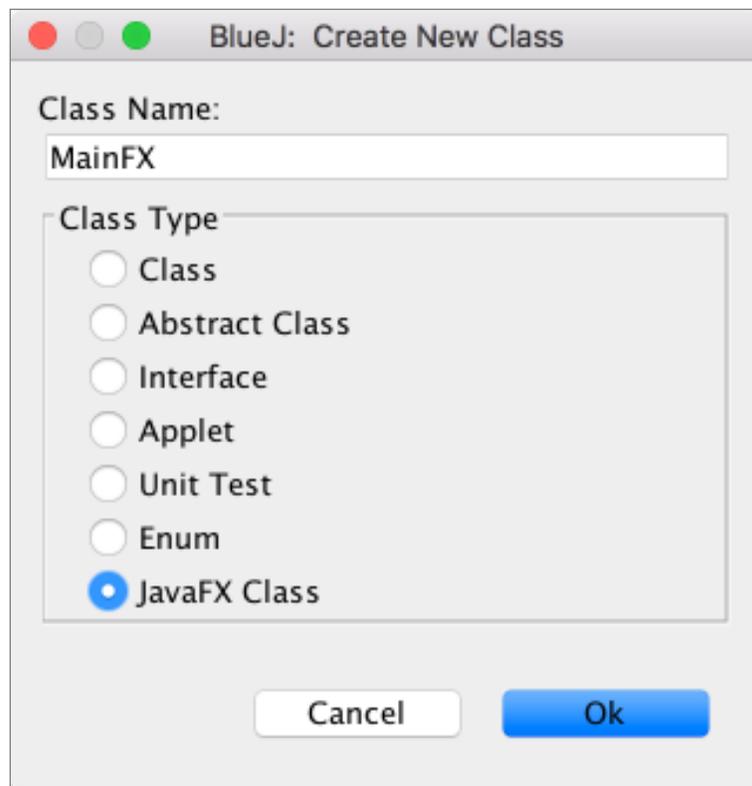
} // end $CLASSNAME
```

The above template is for a JavaFX application. Since blueJ does not quite support JavaFX, this template serves as a work around. I have added my own specific flavor to this to suit the teaching style and conventions I use here at SIUE.

Feel free to add you own style and make it truly your own.

3) Make the template available.

Once the template has been edited, you must make it available as an option in the Create New Class dialog.

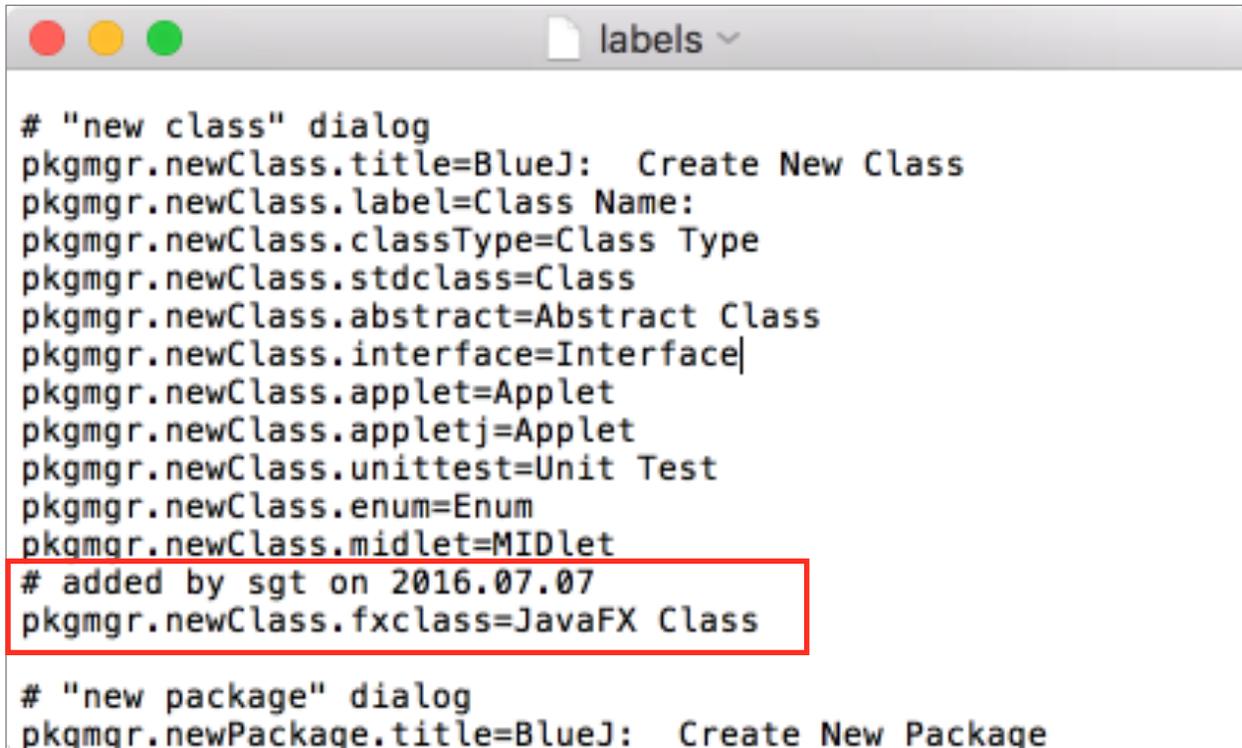


This is done by editing the <blueJ>/lib/<language>/label file.

Note: On a MacOS, this file is the <language> directory along with the templates directory.

Locate the # "new class" dialog section and add your new template's entry.

Since I created a new JavaFX template called `fxclass.tpl`, I added the line `pkgmgr.newClass.fxclass=JavaFX Class`.



```
# "new class" dialog
pkgmgr.newClass.title=BlueJ: Create New Class
pkgmgr.newClass.label=Class Name:
pkgmgr.newClass.classType=Class Type
pkgmgr.newClass.stdclass=Class
pkgmgr.newClass.abstract=Abstract Class
pkgmgr.newClass.interface=Interface|
pkgmgr.newClass.applet=Applet
pkgmgr.newClass.appletj=Applet
pkgmgr.newClass.unittest=Unit Test
pkgmgr.newClass.enum=Enum
pkgmgr.newClass.midlet=MIDlet
# added by sgt on 2016.07.07
pkgmgr.newClass.fxclass=JavaFX Class

# "new package" dialog
pkgmgr.newPackage.title=BlueJ: Create New Package
```

That is basically all that is needed to add your own template. Use this to add/modify any template and create your own library of customized start up code.

Your mission if you so accept it:

Read more about BlueJ's features by visiting <http://www.bluej.org/doc/bluej-ref-manual.pdf>.