

Creating Form Initialization Code

When you finish [building a GUI Designer form](#), at a minimum you should have a class that is bound to the form, and component type fields in the class bound to the various form components.

If the binary class files are specified as the output option, all runtime initialization code is generated in the class file. If your form has a bound class, you will not see the automatically generated initialization code there. In case you have selected Java source code as the output option, the bound class of your form will contain automatically generated `$$$setupUI$$$()` method.

Do not edit manually the GUI Initializer source code! At any compilation, IntelliJ IDEA automatically generates new code in this section, and all your changes will be lost.

Sometimes you might need to provide initialization code of your own. For example, you want a GUI component to be instantiated by a non-default constructor with certain parameters. In this case, IntelliJ IDEA will not generate instantiation of the component, and it is your responsibility to provide the call to constructor in the `createUIComponents()` method. Otherwise, a Null Pointer Exception will be reported. Follow the general technique described below.

To create custom gui initializer source code for a certain component, follow this general procedure

1. Select the desired component.
2. In the Inspector, check the option **Custom Create**.
3. With the component selected, press **F4**, or choose **Jump to Source** on the context menu.
4. In the text editor, locate the method `createUIComponents()`, and type the desired source code. The code in this method will not be removed on compilation.

Example

For example, you would like to provide non-default constructors for the radio buttons 1 and 2, and have GUI Designer create a default constructor for the radio button 3:

```
...
//For the radio buttons 1 and 2, option Custom Create is set to true.
//You write custom constructors for these components
//in the method createUIComponents()
private JRadioButton radioButton1;
private JRadioButton radioButton2;
//For the radio button 3 the default constructor is generated automatically
//in the method $$$setupUI$$$(). The component properties
//specified in the GUI Designer
//are generated as calls to the set* methods in $$$setupUI$$$().
private JRadioButton radioButton3;
...
private void createUIComponents() {
    radioButton1 = new JRadioButton("Custom text 1");
    radioButton2 = new JRadioButton("Custom text 2");
}
...
private void $$$setupUI$$$() {
    createUIComponents();
    ...
    radioButton3 = new JRadioButton();
    radioButton3.setText("Radio Button");
    ...
}
```

See Also

Reference:

- [Inspector](#)

Web Resources:

- [Developer Community](#) 