

Extracting Hard-Coded String Literals

Having [enabled](#) code inspection that highlights hard-coded string literals, you can proceed with extracting these literals into your properties files. For this purpose, IntelliJ IDEA provides special intention action **i18nize hard coded string literal**.

This section considers two possible ways of accessing resource bundle:

- Using the [java.util.ResourceBundle](#) utility class
- Using [custom utility class](#)

To extract a string literal using `java.util.ResourceBundle`

1. Specify resource bundle that will be used to store the extracted literals. In particular, you can add the following statement to your source code:

```
private static ResourceBundle <field name> =  
ResourceBundle.getBundle("<bundle name>");
```

For example:

```
private static ResourceBundle myBundle =  
ResourceBundle.getBundle("com.intellij.fontChooser.FontChooser");
```

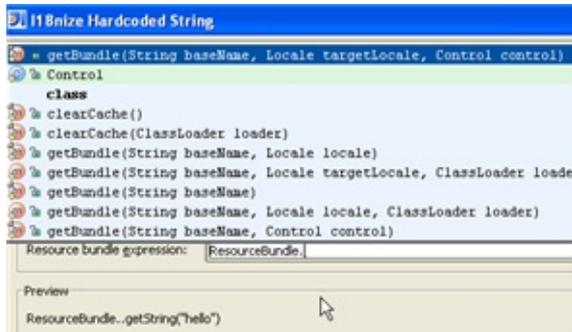
You can skip this step and specify the desired resource bundle expression immediately in the [I18nize Hard Coded String Literal](#) dialog box.

2. Click the highlighted string, press Alt+Enter, and in the list of intention actions choose **i18nize hard coded string literal**:

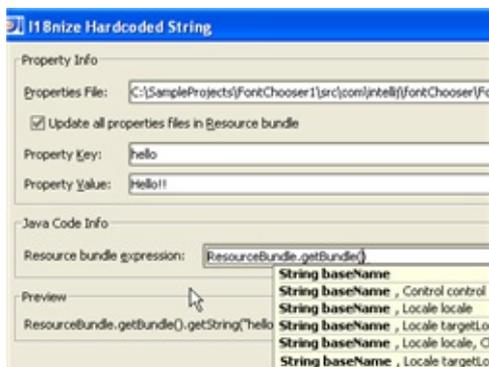


- In the **I18nize Hard Coded String Literal** dialog box specify the *target properties file*, the *property key and value*, and the *resource bundle expression*. If the `ResourceBundle` field has been declared in the source code (as shown in the step 1), IntelliJ IDEA suggests its name by default. If you haven't declared this field in the source code yet, you can still define the desired expression immediately in the dialog box. To do that, enter a valid expression of the `ResourceBundle` type in the **Resource bundle expression** field.

Note that **basic code completion** works in this field. Type class name, and press **Ctrl+Space** after period to select method:



Choose the desired method from the suggestion list, and press **Ctrl+Space** once more to fill in the parentheses:



After that, type the package and resource bundle name in quotes:



- Click **OK**. The line with hard-coded string literal is replaced. For example, if the resource bundle has been declared in the source code, the following line will be created:

```
private String greeting = myBundle.getString("hello");
```

If the resource bundle has been defined in the dialog, the result will be:

```
private String greeting = ResourceBundle.getBundle
{"com.intellij.fontChooser.FontChooser"}.getString("hello");
```

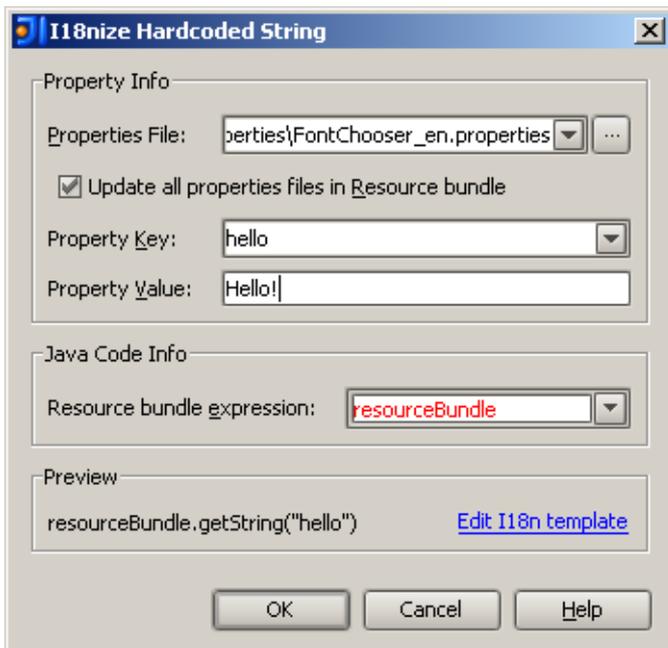
To extract a string literal using custom resource bundle class

- Make sure that `redist/annotations.jar` archive that resides under your IntelliJ IDEA installation, is **added to the module dependencies**.

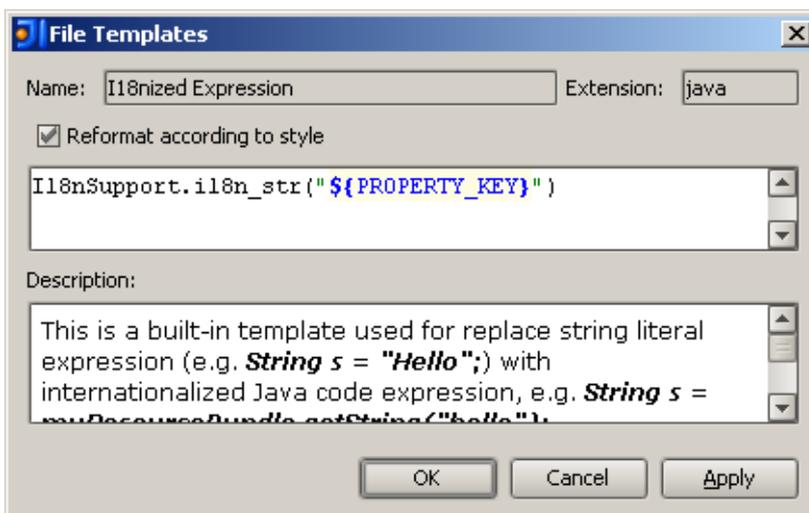
2. Create a new class in your project, and type the following code:

```
import org.jetbrains.annotations.PropertyKey;
import org.jetbrains.annotations.NonNls;
import java.util.ResourceBundle;
import java.text.MessageFormat;
public class I18nSupport {
    @NonNls
    private static final ResourceBundle bundle = ResourceBundle.getBundle ("com.int
public static String i18n_str
    (@PropertyKey(resourceBundle = "com.intellij.FontChooser"))
    String key, Object... params){
    String value = bundle.getString(key);
    if (params.length > 0) return MessageFormat.format(value, params);
    return value;
    }
}
```

3. In a class that contains hard-coded string, click the highlighted string, press Alt+Enter, and in the list of intention actions choose **i18nize hard coded string literal**. The **I18n-ize String Literal** dialog box shows that resource bundle expression is missing:



4. Click **Edit i18n template** link. In the **File Template** dialog box, change the **I18nized Expression** to point to the method of your custom resource bundle class:



Click **OK** to save the updated template and close the dialog box.

This change is global, and affects all projects!

5. In the **Preview** section of the **I18n-ize String Literal** dialog box, see the suggested substitution, and click **OK**. The source code changes:

```
private String greeting = I18nSupport.i18n_str("hello");
```

See Also

Concepts:

- [Internationalization and Localization Support](#)
- [Code Inspection](#)
- [Intention Actions](#)

Reference:

- [I18nize Hard-Coded String](#)

Web Resources:

- [Developer Community](#) 