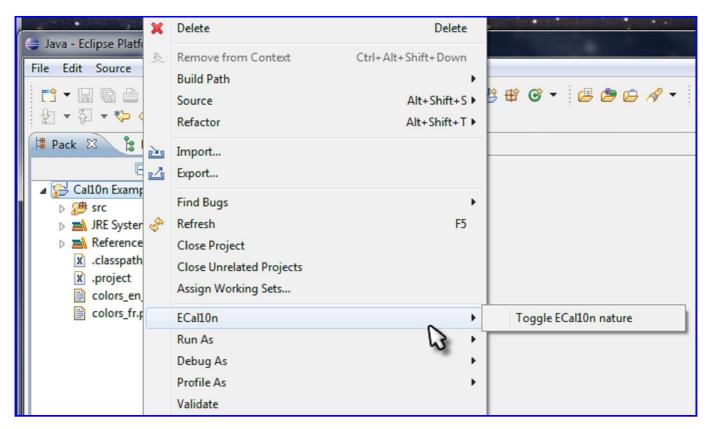# ECal10n

News | Introduction | Usage and screenshots | Installation

I wouldn't think that *all* Java projects in your workspace need localization. So, there's no need for ECal10n to go and parse all the `enum` types it may find – it's fast, and probably not even noticeable, but in my opinion CPU time should be used only when needed.

This is why you'll have to explicitly allow ECal10n to work on your Java project. Quite simple, actually: just right-click on the project and select `ECal10n -> Toggle Ecal10n nature`.

After doing this, any time your sources get compiled ECal10n will:

- parse `enum` types with Cal10n annotations;
- parse only those property files mentioned in the `enum` types as resource bundles (that is, those found starting from the Cal10n `@BaseName` annotation).

Any potential problem is highlighted using markers (visible in the Markers view, available since Eclipse 3.4) and editor annotations (visible in the editor overview ruler and vertical ruler).

For each one, fixes are suggested through the quick-fix keyboard shortcut (`Ctrl-1`), depending on the problem detected.

Missing resource bundle: an `enum` type was annotated with a resource bundle base name, but the corresponding property file was not found. Fix: create the property file.

Empty resource bundle: the resource bundle declared in an `enum` type was found, but contain no key/values whatsoever. Fix: add the `enum` type members' names as keys with empty values.

Empty `enum` type: the parsed `enum` type is tied to a property file containing key/value pairs, but the `enum` type itself has no members. Fix: fill the `enum` type with members, using the property file key names.

Missing item in bundle: the parsed `enum` type has a member whose corresponding key was not found in the relevant property file. Fix: add the `enum` type's member to the property file, as a key with an empty value.

Missing item in `enum`: the parsed property file has a key not found in the declaring `enum` type. Fixes:

- suggest alternative key names based on the `enum` type members' names, using the Levenshtein distance algorithm to find suitable candidates;
- remove the key/value pair from the property file;
- add the property name to the `enum` type's members.

```
Colors.java ⊠
 1  package com.example;
 2
 3⊕ import ch.qos.cal10n.BaseName;▯
 6
 7  @BaseName("colors")
 8  @LocaleData(
 9          defaultCharset="UTF8",
10          value = {   @Locale("fr") }
11  )
12  public enum Colors {
13      RED,
14      GREEN,
15      BLUE,
16      BLUISH
17  }
18
```

```
colors_fr.properties ⊠
1 RED=les roses sont  rouges
2 GREEN=les {0} sont verts
3 BLUE=les violettes sont bleues
4 BLUIE=les violettes sont bleues
5
```

⇨ Change to BLUE
⇨ Change to BLUISH
✖ Delete this line

The current key does not seem to exist in the related enum class, but this similar one does. Select this item to use it.

Press 'Tab' from proposal table or click for focus