i-net Clear Reports 2017

Internationalization Guide

i-net Designer

i-net software

1 Introduction

When creating reports, many different aspects must be taken into account. One important point is the versatility and re-usability of a report. Especially in cases where reports will be viewed by many users which may be located in different countries and so speak different languages. Creating an individual report for each language would take a considerable amount of effort. And even if these reports were created, any changes made to one report would have to be applied to each localized version of a report.

To avoid this, i-net Clear Reports offers the opportunity to develop reports which support localization. The report is localized at run-time using the locale of the user requesting the report.

Localization of reports can be done quite easily. It is both possible to localize a single individual report, as well as to provide a global set of translations which will be applied to all reports provided by the server. This guide will show which parts of a report can be localized and the ways this localization can be accomplished.

2 How are reports localized?

When a report is rendered, each translatable bit of text content in the report is checked for whether a translation exists for the locale of the user requesting the report. Please see chapter "Which parts of a report are localized?" for exactly which content is translatable.

For example, if your server or i-net Designer is running with an English locale, and a user with a French locale requests a report, the report's various texts are checked for French translations. If they exist, they are replaced, giving your French user the report with French labels and a French locale for the numbers and formats in the report. If the French translations does not exists, then the translations for the default language (locale) are used.

The default language is either the language of the machine running the report server or i-net Designer (see: Change the system locale) or it is the language that was specified as start parameter of report server or designer by setting the system properties `user.language` and `user.country`.

With the following command you can start the i-net Designer with the German locale on a system with English or any other locale:

```
java -Duser.language=de -jar core/Designer.jar
```

To start the report server with another locale you need to modify the start script of the server. Which script you have to modify depends on the environment in that you have installed the i-net Clear Reports report server.

The translation is done by first checking for individual translations stored directly in the report being created. Then, if the report is connected to a specific resource bundle, this bundle is checked for a translation. If still no translation is found, the list of global translations which is installed server-wide and is used for all reports is checked.

2.1 Individual (report-based) translations

These translations are stored within the report itself and only are applied to the given report. Only if no report-based translation can be found for a specific label, are the global translations employed for a report. There are two ways to provide report-based translations.

2.1.1 Via i-net Designer

Using the translation dialog it is possible to provide translations for your report from within the i-net Designer itself. You can access the translations dialog through the menu item "Report | Translations..."

The dialog "Translation Settings" appears. In this dialog you can manage which languages you are offering for your report. You can add, remove, and export your translations from here. See the i-net Designer help for more information on the various options.

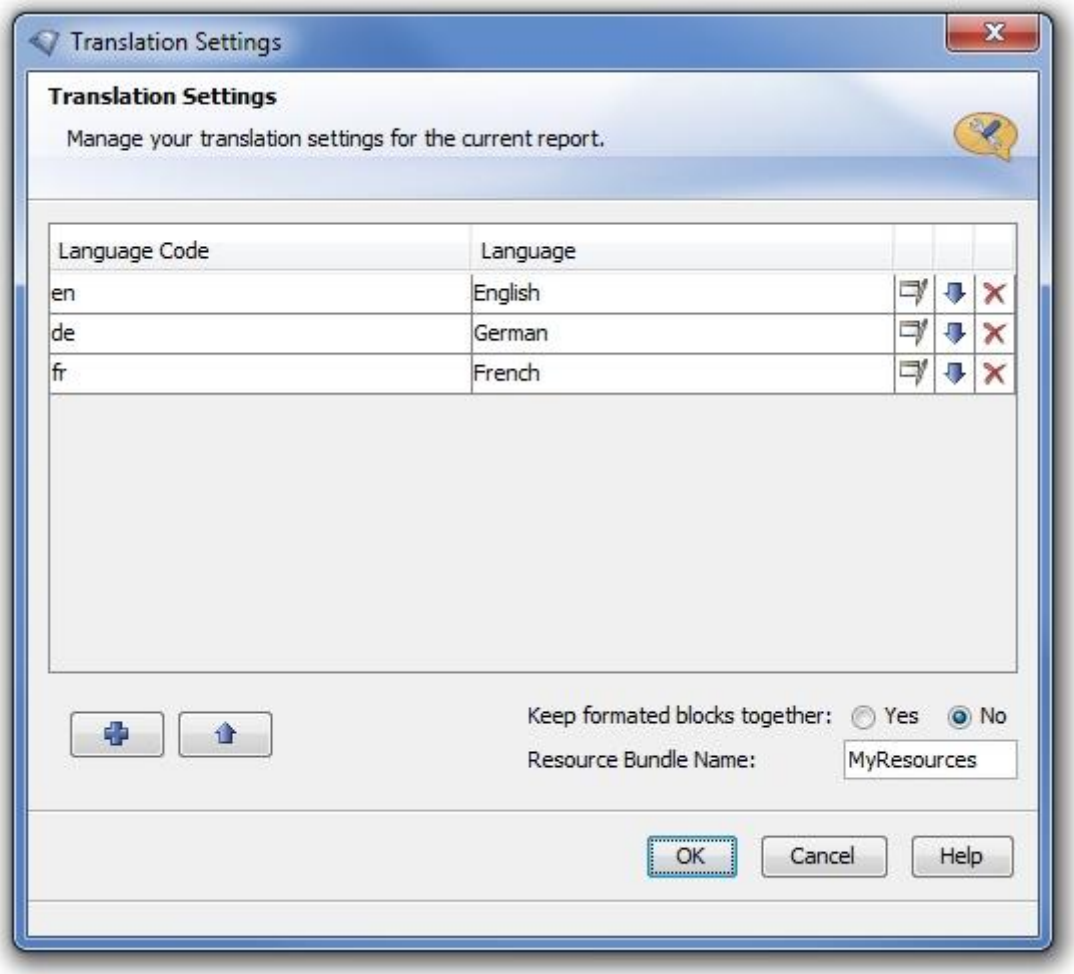

Figure 1: Translation settings Dialog

A click on the edit icon will enable you to actually enter the translations themselves for the language you chose in the language editor. The language editor will display all known labels and text content in the report which can be translated.

- Red lines are text for which there is currently no translation. These are the lines you need to provide a value for.
- Blue means the report value is no longer found in the report (and is most likely outdated) – deleting the translation will remove this entry.
- White means there is a translation and it will be used to replace any text in the report which equals the "report value".

It can be helpful to have a "reference language" for translating new labels, since most labels you want to translate will be keys instead of full words or phrases (see Best Practices in chapter "Best Practices for Localization").

Figure 2: Language Editor

2.1.2 Via API

If you are making use of the powerful i-net Clear Reports API, you can also provide translations programmatically. Here is a quick and simple sample for adding report-based translations for a report you have:

2.1.2.1 Java sample code

```
Engine engine = RDC.loadEngine(new File("myReportFile.rpt"));
Translations t = engine.getTranslations();
// Create the English translations as a Properties
```

```
Properties engprops = new Properties();
engprops.setProperty("reportTitle", "My Report Title");
...
// Create the German translations as a Properties
Properties gerprops = new Properties();
gerprops.setProperty("reportTitle", "Mein Berichtstitel");
...
t.setTranslation(Locale.ENGLISH, engprops);
t.setTranslation(Locale.GERMAN, gerprops);
```

2.1.2.2 .NET sample code

```
Engine engine = RDC.LoadEngine(new java.io.File("myReportFile.rpt"));
Translations t = engine.GetTranslations();
// Create the English translations as a Properties
Properties engprops = new Properties();
engprops.setProperty("reportTitle", "My Report Title");
// Create the German translations as a Properties
Properties gerprops = new Properties();
gerprops.setProperty("reportTitle", "Mein Berichtstitel");
...
t.SetTranslation(Locale.ENGLISH, engprops);
t.SetTranslation(Locale.GERMAN, gerprops);
```

2.2 Resource Bundles for multiple reports

If you have multiple reports with a similar topic, it could easily happen that your reports will have identical labels. In order to not have to re-translate the labels for each individual report, it is possible to create a resource bundle and point your reports to it. These reports will then take the resource bundle for translations.

Note that this is not the same as specifying a global resource bundle for all reports run on the same server. Rather this resource bundle will only be used for the reports which point to it.

For more information on resource bundles, see the Java documentation to ResourceBundle at: http://download.oracle.com/javase/7/docs/api/java/util/ResourceBundle.html

In order to use a resource bundle for multiple reports, your resource bundle must simply be included in the class path. You can do this by either moving it into the lib directory (configurable in the i-net Clear Reports configuration) or by modifying the classpath VM parameter when starting the server or the designer.

2.2.1 Via i-net Designer

For each report you wish to connect to the resource bundle (which must have been placed in the classpath of the running server or designer), you enter the resource bundle's name in the Translation Settings dialog.

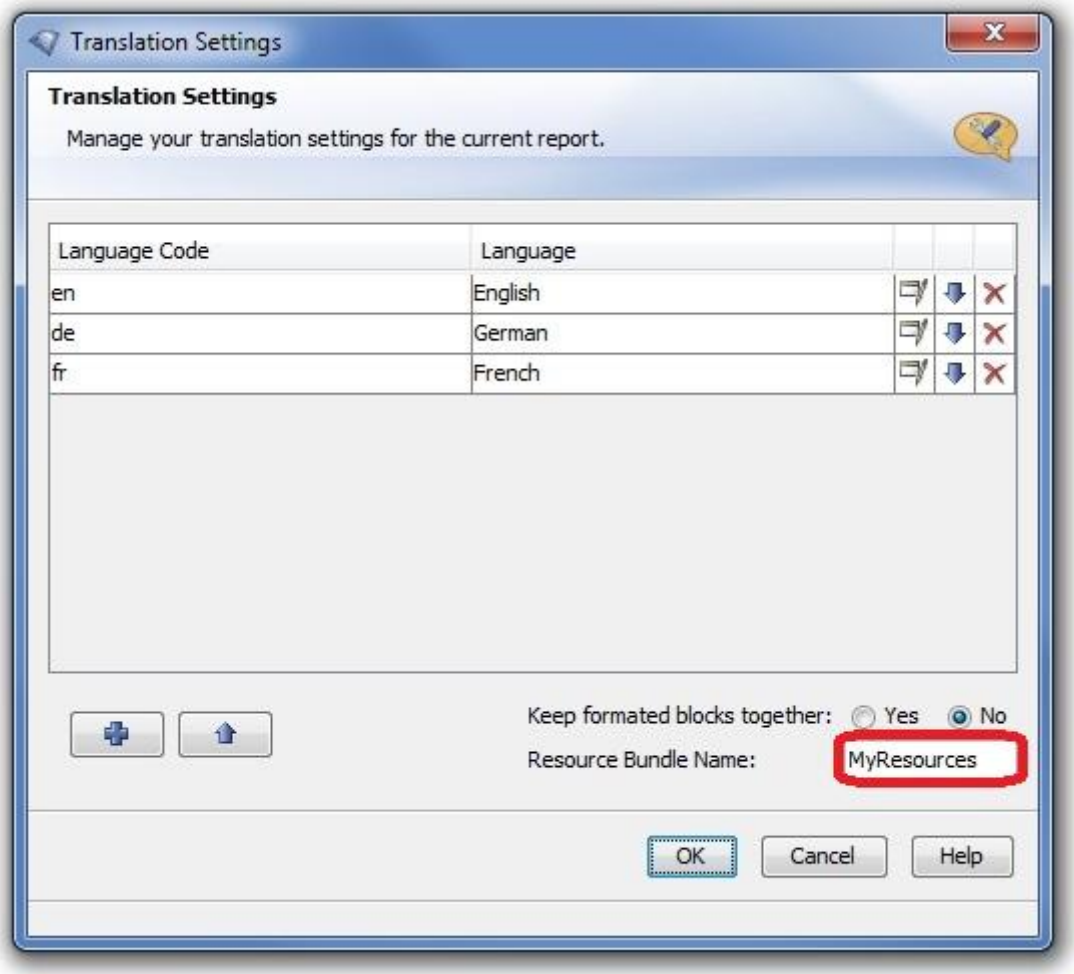

Figure 3: Specifying a resource bundle

2.2.2 Via API

Here is a quick and simple sample for setting the resource bundle for a certain report:

2.2.2.1 Java sample code

```
Engine engine = RDC.loadEngine(new java.io.File("myReportFile.rpt"));
Translations t = engine.getTranslations();
t.setResourceBundleName("MyResources");
```

2.2.2.2 .NET sample code

```
Engine engine = RDC.LoadEngine(new java.io.File("myReportFile.rpt"));
```

```
Translations t = engine.GetTranslations();
t.ResourceBundleName = "MyResources";
```

2.3 Global Resource Bundle for all reports

For labels which will be the same for all your reports, you may want to specify a global resource bundle for your server. These translations are used for labels if no local report-based translation is found for them and the report's resource bundle (if the report points to one) does not contain a translation either.

For more information on resource bundles, see the Java documentation to ResourceBundle here: http://download.oracle.com/javase/6/docs/api/java/util/ResourceBundle.html

2.3.1 Providing your own Resource Bundles

In order to globally register a resource bundle, your resource bundle must first be included in the class path. Do do it, add the resource bundle files to a JAR file (using jar command of the Java VM) and add this to the class path by either moving it into the lib directory (configurable in the i-net Clear Reports configuration) or by modifying the classpath VM parameter when starting the server or the designer.

You then need to configure your server's LanguageResources setting. You can do this either in the Swing-based configuration manager or in the web-based remote configuration manager which you can reach in your browser (e.g. http://<server>:9000/remote). More information about the usage of the configuration manager you can find in the documentation.

Simply enter the name of your global resource bundle in the "Language Resource" setting (found in the category "Customization").

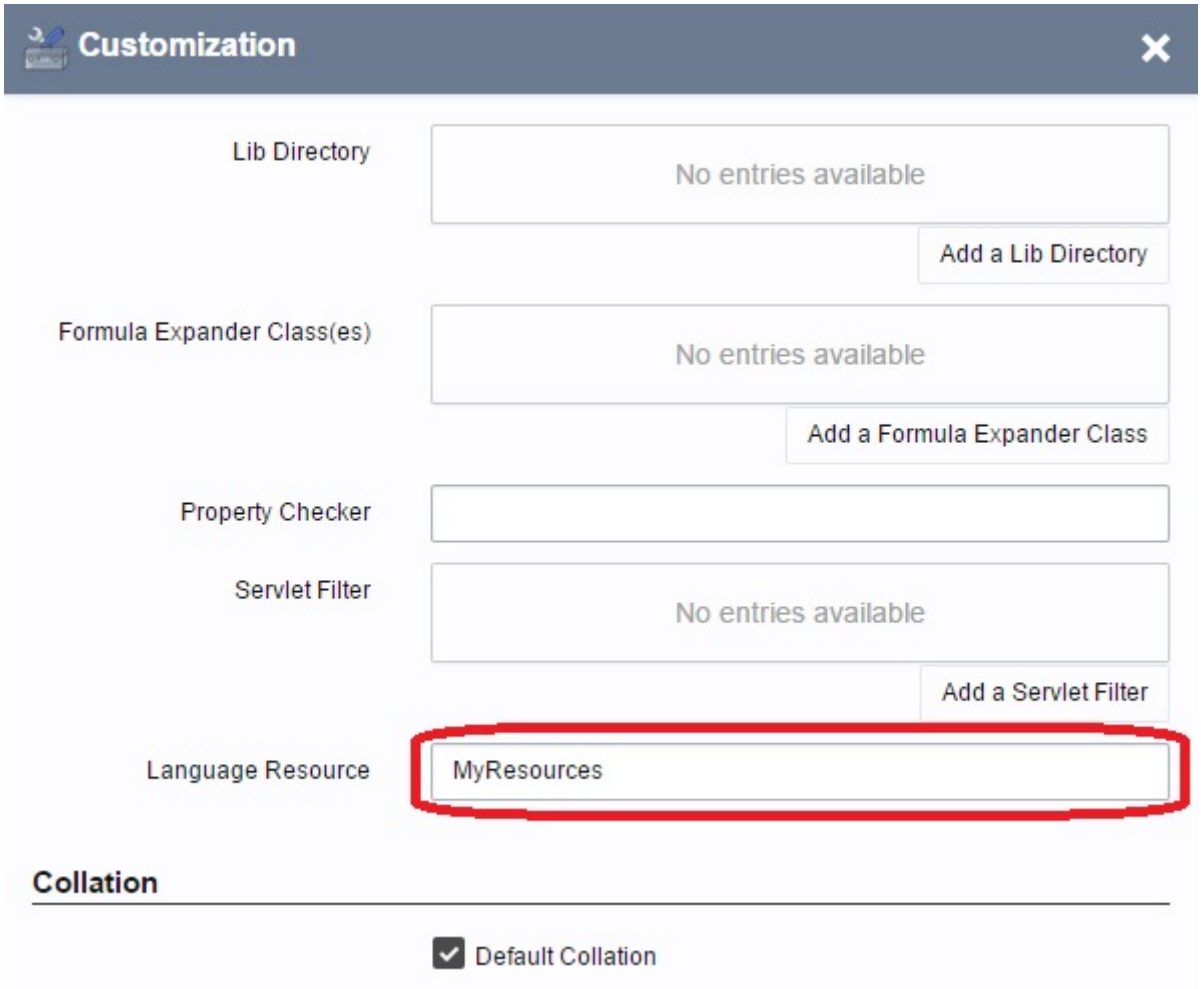

Figure 4: Configuring Global Language Resources

3 Which parts of a report are localized?

Virtually all text content of a report can be localized. Here are the elements that are attempted to be translated:

- **Labels (text elements)**
  - Note: that the entire label's text is taken as the translation key (not the individual words), unless there are FieldParts in the label, in which case each text part by itself is translated.
- **Report Title**
- **Chart Labels**, i.e. the title, sub-title, footnote, and axis titles of charts
- **Prompt names** - when the user is prompted for parameter values, the names are translated, however the report prompt names used by the API, etc., remain the same.
- **Formula function** `translate()` - The formula function translate(String) attempts to find a translation for its parameter it passes through. This is the only way to translate dynamic content – all other labels are "static", i.e. set when designing the report. Using a formula such as `translate( {Table.DatabaseField} )` will cause all values of this field to be translated.

4 Best Practices for Localization

4.1 Use keys instead of full words

Using keys such as "Header.title" instead of real English labels such as "Yearly Report" has the advantages that it is harder to overlook missing labels in other languages, also that it is easier to change your labels by simply changing the translations rather than the more abstract key names.

4.2 Use the appropriate scope for translation

In order to avoid unnecessary work, you should make use of the global Language Resources whenever possible. This will enable you to re-use translations throughout your reports.

If it looks like you may have a group of similar reports, it may make sense to export a single report's translations and use it as a basis for a multi-report resource bundle.

4.3 Test your translations

Translations are the source of the common problems or slight mistakes. We'd recommend running a "staging" server or using the Designer's remote repository functionality to preview your reports in various languages before you publish them on your production server.

4.4 Pay attention to format differences

In addition to your translation, you will also want to format your fields in a way that locale is respected. For example, the number "1.000" in an English locale is the number 1, while in a German

locale, it is the number "1000". If your report is going to be viewed internationally, we'd recommend therefore to use the "default format" for your data fields if possible.

5 Further Resources

At most places within i-net Designer, pressing F1 will bring up the online help document. You can get it also with the menu item "Help". The online help is searchable, has an index and a table of contents. This should explain the answers to many of the questions you might have.

Additionally, there is an extensive FAQ on our web site, located at: http://www.inetsoftware.de/documentation/clear-reports/online-help/support/faq/index.html.

Also, there are various guides located on our web site which are similar to this one, on different topics such as the Visual Database Wizard, Grouping and Sorting - if you should still have questions or problems not answered or solved in these resources, you can send an email to the i-net Clear Reports support team at: clearreports@inetsoftware.de.