

Resource bundles

This document describes how to use resource bundles. These resource bundles are only used inside the xReporter server. The presentation layer uses another mechanism, the Cocoon i18n transformer.

Resource bundles contain language-dependent text strings. The resource bundles can be divided into two groups:

- built-in resource bundles: for default xReporter messages. At the time of this writing, these are available in English and Dutch (Belgium). These bundles should normally not be edited by the user (except for making corrections or adding a new language).
- "user bundles": for texts used in report definitions, datatypes and datasources. For example the description of a report, the prompt of a datatype, ...

Usually you will only need to work with the user bundles. There are three distinct user bundles:

- report: for all texts used in report definitions
- datatype: for all texts used in datatypes
- datasource: for all texts used in datasources

The actual filenames for the resource bundles can be configured in config.xml (inside the resourcemanager element). By default it is configured so that the report and datatype user bundles refer to the same file. This is because otherwise it can be confusing from which bundle a text is taken (since datatypes can also be declared inline in report definitions).

For each language (or "locale" to be exact) and each resource bundle there is a different file. Let's suppose that the file for the report-userbundle has the basename "ReportMessages". Then the filename for the 'Belgium-specific' Dutch translation would be ReportMessages_nl_BE.properties. If this file cannot be found, a file called ReportMessages_nl.properties (holding the common wording for all country-variants of the Dutch language) will be used, and if that cannot be found, the file for the default system locale will be searched for, and if finally that one cannot be found, the file ReportMessages.properties will be used.

See also the example resource bundles of xReporter.

Starting from xReporter 1.3, you can have multiple files for the same user resource bundle. This means you can split the bundles into multiple files, and xReporter will search them one by one (and then do locale-fallback and search them again one by one). Configuring multiple bundles is done in the config.xml, by having multiple <userbundle> elements with the same name but different dir or base-name attributes:

```
<userbundle name="report" dir="@conf.home@/resourcebundles" base-name="ReportMessages" />
<userbundle name="report" dir="@conf.home@/resourcebundles" base-name="ReportMessages2" />
```

WARNING

(From xReporter 1.3 on, the following does not apply anymore to the user bundles, only to the built-in application bundles) While the above statement just explains the reality of the used ResourceBundles, we'd like to put some stress on the fact that the mentioned *default system locale* will be the one of the server-side system running the xreporter engine (phoenix). This locale cannot be in any way relevant to what the end-user actually is expecting. In server applications like xReporter the described fallback mechanism tends to lead to unexpected behaviour that varies with the locale on the server (which might be changed outside the powers of the xreporter-admin) or the chosen languages in the baseName.properties (the ones declaring no language-suffix). The advice therefore is: (1) do not provide catch all translations in properties files without language-specifications (remember there is no suitable language you could pick for that file) and (2) always provide translations for all keys (using the fall-back mechanism only between country-variants).

NOTE

xReporter end-users are encouraged to provide other than the current translations for the various resourceBundles in the distribution. Please find them by searching for all *.properties in the ./src folder. Just send your translations (or fixes to current ones) to the xReporter mailing list.