Localization
Better, faster, easier through sharing resources

In 2001, a 30+ team of programmers was to develop a large application (it still is under development) that would replace more than 20 legacy applications.

Phase 1 deliverables had a fixed mandatory date, and time was running fast.

Translation Bureau could really not deliver in only one official language and the translation later. Official Languages Act makes it mandatory to deliver documentation and software simultaneously in English and French. So it really had to be delivered simultaneously in both official languages.

Back in 2001, some people were proud to say that they could take "only" one month to translate rather than six or nine as soon as the development would stop. In our case, the solution was to translate **during the development**, and to maintain terminology and definitions from day 1. Developers could consult language specialists to maintain a better terminology, translators could consult programmers to do a better translation, and the process was seamless.

From the moment where programmers stopped to work on phase 1, **12 hours later** the application was delivered in both official languages. This was the result of a collaborative effort involving programmers and translators and using a terminology management tool as a common ground.

Developers would all textual resources (button, menus, error messages, etc.) in the terminology management software and indicate what linguistic process should be done (translate, revise or translate and revise). A few lines of programming allowed one to generate the interface whenever required.

Since start, there were conventions to respect the rules of internationalization that applied (leave enough space for French that is longer than English, never merge messages, use specific rather than generic messages, etc.) In addition, they could provide the translators items of info that are lacking in most localization jobs.

Programmers could also search for existing items rather than reinventing the wheel. Various search methods allowed to find rapidly. This would avoid the translators to call programmers so that they can tell what is the difference between two almost identical texts (when there was none).

The phone numbers of authors were automatically logged, so that they could be reached if any clarifications were still needed. But translators really did not have to call often.

Sometimes, the programmer was bilingual, he would then write directly a name for the button or his message in both languages, but would then check the option Revise. Otherwise, he would check Revise and translate. Some programmers know it is a good idea to have their text revised, even in their mother tongue.

The translators could easily find the records that needed to be translated or revised, and indicate what had been done (revision, translation or revision and translation).

There was a full-time writer for the use cases and other documentation. He could also search in the terminology database and make sure his terminology was consistent, and write new terms.

Translation of those documents was done also on a continuous basis, using a translation memory to speed-up things.

The translation effort required was a bit bigger than on a normal project, but there was almost no delay due to translation.

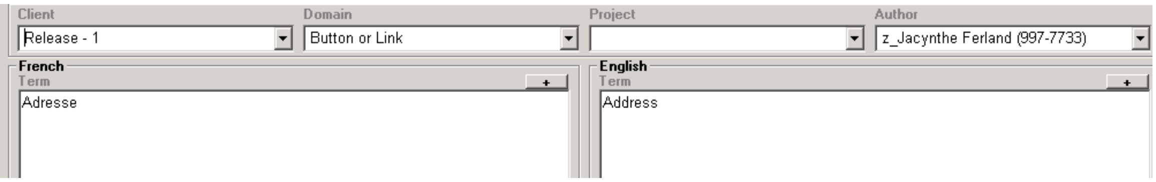
We can safely say that the additional costs of translation certainly did not exceed the savings on idle time on the programmer's side.

The fact that the application was translated in a continuous basis also allowed for bilingual parallel quality testing (the translators could see the application in English and French in test environment).

Now I will be looking at Wikis to see how they could also help us especially for multilingual documentation and terminology in the future.

Some screen captures follow.

André Guyon
Translation Bureau / Bureau de la traduction
Public Works and Government Services Canada/ Travaux publics et Services gouvernementaux Canada



Client: Release - 1

Domain: Button or Link

Project:

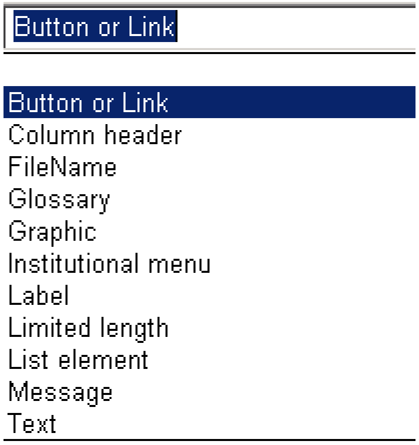
Author: z_Jacynthe Ferland (997-7733)

French

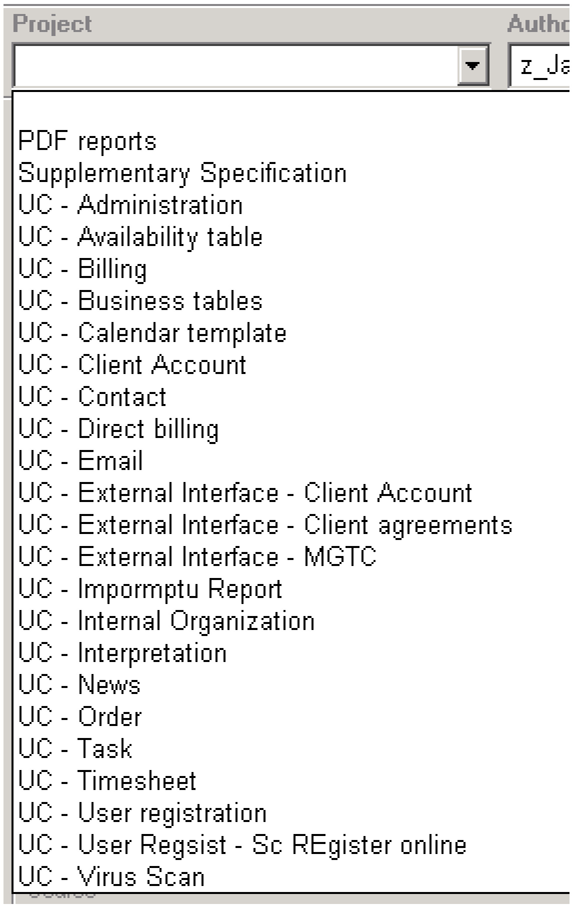
Term: Adresse

English

Term: Address

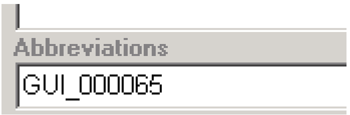


Button or Link

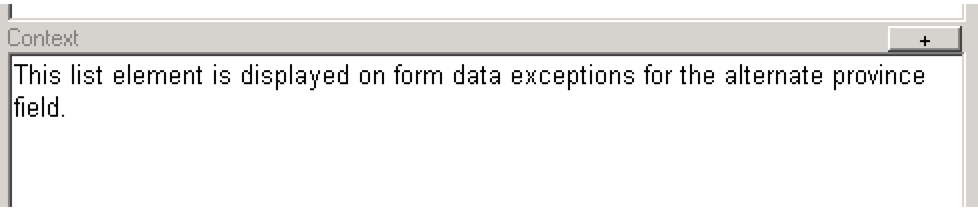


PDF reports
Supplementary Specification
UC - Administration
UC - Availability table
UC - Billing
UC - Business tables
UC - Calendar template
UC - Client Account
UC - Contact
UC - Direct billing
UC - Email
UC - External Interface - Client Account
UC - External Interface - Client agreements
UC - External Interface - MGTC
UC - Impormptu Report
UC - Internal Organization
UC - Interpretation
UC - News
UC - Order
UC - Task
UC - Timesheet
UC - User registration
UC - User Regsist - Sc REgister online
UC - Virus Scan

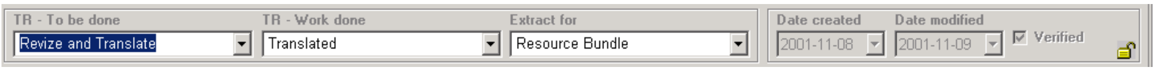
Instead of an Abbreviation, we have the ID of the string



Abbreviations
GUI_000065



Context
This list element is displayed on form data exceptions for the alternate province field.



TR - To be done: Revize and Translate

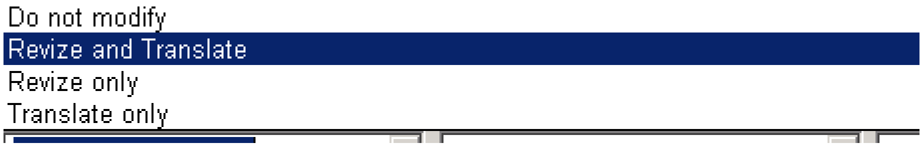
TR - Work done: Translated

Extract for: Resource Bundle

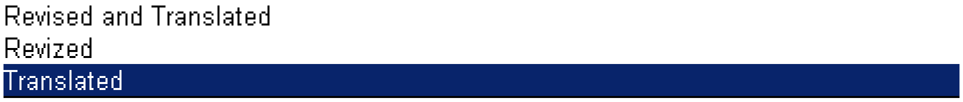
Date created: 2001-11-08

Date modified: 2001-11-09

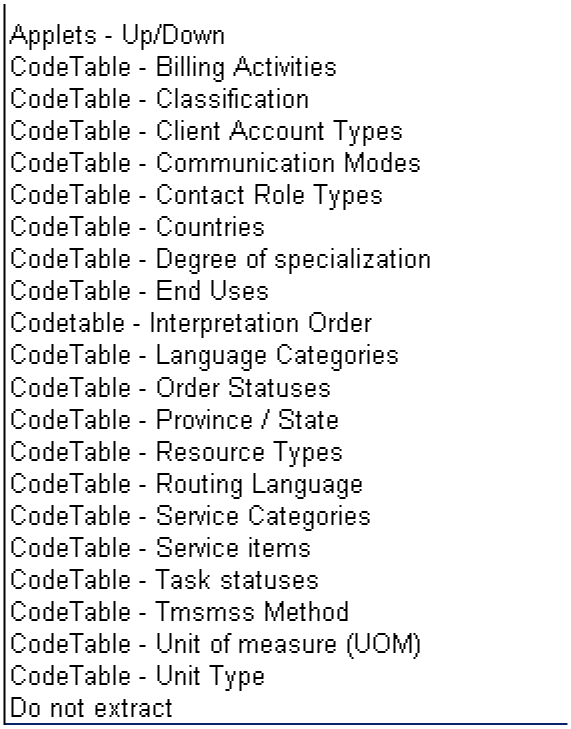
☑ Verified



Do not modify
Revize and Translate
Revize only
Translate only



Revised and Translated
Revized
Translated



Applets - Up/Down
CodeTable - Billing Activities
CodeTable - Classification
CodeTable - Client Account Types
CodeTable - Communication Modes
CodeTable - Contact Role Types
CodeTable - Countries
CodeTable - Degree of specialization
CodeTable - End Uses
Codetable - Interpretation Order
CodeTable - Language Categories
CodeTable - Order Statuses
CodeTable - Province / State
CodeTable - Resource Types
CodeTable - Routing Language
CodeTable - Service Categories
CodeTable - Service items
CodeTable - Task statuses
CodeTable - Tmsmss Method
CodeTable - Unit of measure (UOM)
CodeTable - Unit Type
Do not extract