

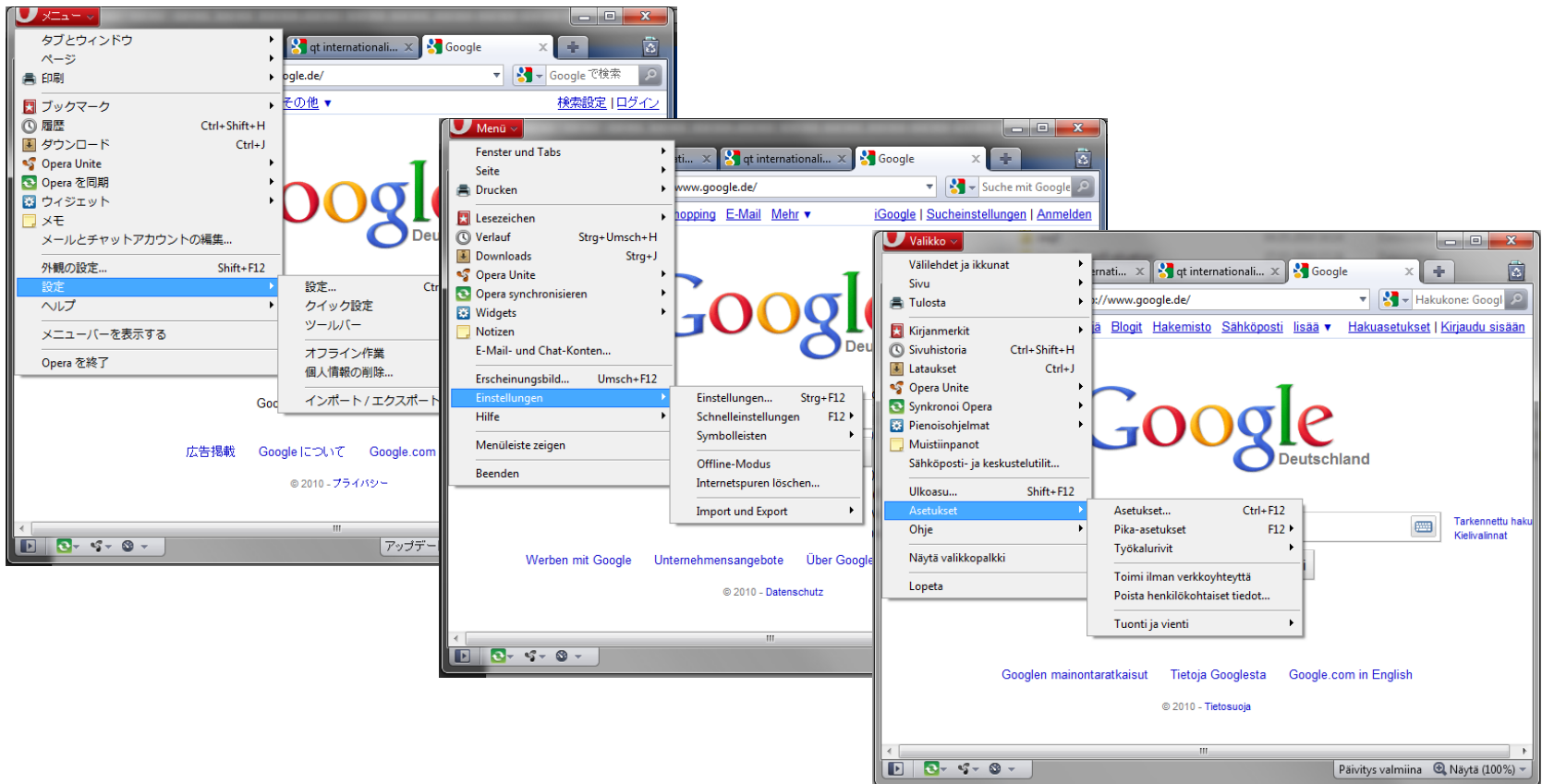
9/8/2010

Internationalization using QT

Markus Engel

What does Internationalization mean?

“The internationalization of an application is the process of making the application usable by people in countries other than one's own.”



What to consider in Internationalization?

- Language and Alphabet
- Character encodings
- Presentation Conventions (ligatures)
- Input techniques
- Time and Date

Translation in 3 steps:

1. Extraction of translatable text from the source code. (*lupdate*)
2. Offering translations for each text. (*Qt-Linguist*)
3. Create binary file that is used by the application. (*lrelease*)

- Use QString for all user visible texts
 - Unicode → support for every language

- Mark them as translatable using *QObject::tr()*

```
QLabel *labelThresholdMin = new QLabel(tr("Lower Threshold:"),  
                                       controlsParent);
```

```
QLabel *labelThresholdMax = new QLabel(tr("Upper Threshold:"),  
                                       controlsParent);
```

```
QPushButton* button = new QPushButton(tr("start region growing"),  
                                       controlsParent);
```

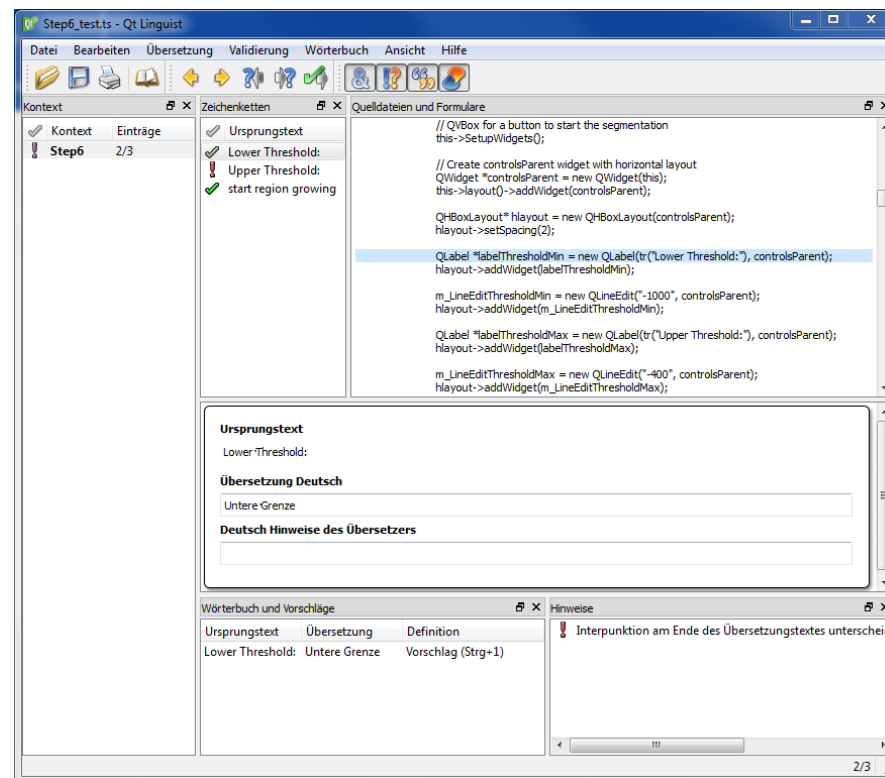
- Tell the application which translation to use

```
QApplication qApp( argc, argv );  
QTranslator translator;  
translator.load( "Step6_DE.ts" );  
qApp.installTranslator( &translator );
```

- Create translation-file

```
lupdate Step6.cpp -ts Step6_DE.ts
```

- Define translations using Qt Liguist



- For simple projects:

```
# Translation files
SET(TRANS Step6_test.ts)
# add translations ...
QT4_ADD_TRANSLATION(QM ${TRANS})
[...]
SET(STEP6_CPP ${STEP6_CPP} ${STEP6_MOC_CPP} ${QM})
```

- Internationalization with QT:

<http://doc.trolltech.com/4.6/internationalization.html>

- QT Linguist Manual:

<http://doc.trolltech.com/4.6/linguist-manual.html>

- Integration into CMake:

[http://www.cmake.org/Wiki/CMake:How To Build Qt4 Software](http://www.cmake.org/Wiki/CMake:How_To_Build_Qt4_Software)