

## 11.4 Internationalisierung von Projekten (I18N)

Da das Online-Gambas-Buch nur in deutscher Sprache vorliegt, werden die Projekte zu den einzelnen Kapiteln seit zwei Jahren in englischer Sprache (Standard) mit deutscher Übersetzung entwickelt. Damit erhalten die Leser Projekte, die in Deutschland in deutscher Sprache genutzt werden können. Alle anderen Leser verwenden das Projekt jedoch in der englischen Version, da keine weiteren Übersetzungen angefertigt werden.

### 11.4.1 Wie konfiguriert man ein Projekt, das übersetzt werden soll?

Schon beim Anlegen eines neuen Projekts können Sie festlegen, ob das Projekt übersetzt werden kann. Nachdem ein neues Projekt angelegt wurde, gehen Sie in das Menü Projekt > Eigenschaften > Optionen. Dort können Sie prüfen, ob in der Rubrik 'Übersetzung' 'Ja' angegeben ist, damit das Projekt übersetzt werden kann. Wählen Sie die Standard-Sprache 'Englisch'. Zu empfehlen ist für die englische Sprache die Lokalisation 'Englisch (allgemein)'. Das hat den bereits erwähnten Vorteil, dass immer dann, wenn keine Sprachdatei für die aktuellen Gebietsschemaparameter (Locale) existiert, das Programm auf die Standard-Sprache Englisch zurückfällt, damit auch Leser aus anderen Ländern zumindest die den Kapiteln beigelegten Projekte in der englischen Version testen können.

Nach der Änderung der Projekteigenschaften erscheint in der Toolbar der IDE ein neuer Button mit einer kleinen blauen Flagge, der die Projektübersetzung (Dialog) öffnet. Mit CTRL+T rufen Sie die Projektübersetzung direkt auf. Dann sehen Sie im Fenster 'Projektübersetzung' in der Combo-Box neben der blauen Flagge als Symbol für einen Menü-Button nur den Eintrag (Standard) und rechts neben dem grünen Pfeil den Eintrag 'Deutsch (Deutschland)' in der ComboBox. Das mag Sie am Anfang irritieren – ist aber korrekt, denn die in den Projekteigenschaften eingetragene Vorgabesprache ist Englisch und wird nicht extra herausgestellt! Der Eintrag 'Deutsch (Deutschland)' erscheint, weil im Falle des Autors die Gebietsschemaparameter für Deutschland gesetzt sind. Später können Sie mit dem Eintrag 'Neu' in den Einträgen des Menü-Buttons weitere Übersetzungen anlegen und bestehende modifizieren.

Wenn ein übersetzbares Gambas-Programm zum Beispiel mit der Standardsprache Englisch startet, dann erkundet der Interpreter die aktuelle Locale. Ist für die aktuelle Locale eine Übersetzung in einer Sprachdatei vorhanden, so wird diese geladen. Die Übersetzung liegt zum Beispiel für die deutsche Version eines Gambas-Projektes in der Sprachdatei de.po im (versteckten) Projekt-Ordner .lang. Hier sehen Sie einen Ausschnitt aus einer Sprachdatei de.po, aus dem der Dateiname, die Zeilennummer sowie die originale englische Zeichenkette und deren deutsche Übersetzung erkennbar sind. Beachten Sie, dass die letzte Zeichenkette "Spine (Carbon):" nicht übersetzt wurde:

```
...
#: FMain.class:11
msgid "Spine Value Calculator for Arrows"
msgstr "Spinewert-Berechnungen für Pfeile"

#: FMain.form:88
msgid "Spine (Wood):"
msgstr "Spine (Holz)"

#: FMain.form:104
msgid "Spine (Carbon):"
msgstr "-"
```

Ansonsten wird die in den o.a. Optionen angegebene Standard-Übersetzung geladen. Das erkennen Sie daran, dass stets die englischen Zeichenketten angezeigt werden, so lange Sie diese noch nicht übersetzt haben. Daraus folgt auch, dass Sie die Übersetzung schrittweise vornehmen können. Beachten Sie, dass eine Änderung der (englischen) Original-Zeichenketten immer eine neue Übersetzung erfordert, weil diese automatisch gelöscht wurde! Die Übersetzungen werden *automatisch* gespeichert, wenn Sie die Projektübersetzung schließen.

### 11.4.2 Wie kennzeichnet man zu übersetzende Texte?

Alle Texte, die auf der grafischen Benutzeroberfläche (GUI) zu sehen sind, werden automatisch als zu übersetzender Text in der Projektübersetzung gekennzeichnet. Im Programm-Quelltext müssen Sie übersetzbare Zeichenketten in runde Klammern setzen, um sie als zu übersetzenden Text auszuzeichnen. Erst dann werden diese von der Projektübersetzung erkannt und es wird im Dialog eine Übersetzung eingefordert. Dabei müssen Sie sich stets Gedanken machen, ob es sinnvoll und notwendig ist, einen String zu übersetzen.

### 1. Fall

```
blSpineCarbon.Text = "Spine (Carbon):"  
blSpineCarbon.Text = ("Spine (Carbon):")
```

In diesem Fall sind beide Schreibweisen möglich, denn der Text muss nicht notwendigerweise übersetzt werden. Für die erste Zeile ist nichts zu übersetzen, die Zeichenkette erscheint nicht in der Projektübersetzung. Für die zweite Zeile wird als Übersetzung ein Bindestrich - der String "-" eingetragen. Das bedeutet, dass der String "Spine (Carbon):" als zu übersetzender Text beachtet wurde, aber keiner Übersetzung bedarf bzw. nicht übersetzt wird – die Original-Zeichenkette "Spine (Carbon):" wird angezeigt.

### 2. Fall

```
' FMain.Caption = "Spine Value Calculator for Arrows"  
FMain.Caption = ("Spine Value Calculator for Arrows")
```

Die komplette Zeichenkette "Spine Value Calculator for Arrows" wird durch die beiden runden Klammern als 'zu übersetzen' gekennzeichnet.

### 3. Fall

```
' Message.Error("The value for the tensile weight is missing!")  
Message.Error(("The value for the tensile weight is missing!"))
```

Die komplette Zeichenkette "The value for the tensile weight is missing!" als Argument der Methode Message() ist zu übersetzen.

### 4. Fall

Bei der folgenden Zeile, die einen Filter festlegt

```
Dialog.Filter = ["*.png;*.jpg", "Image files"]
```

würden Sie nur den zweiten String "Image files" übersetzen, denn der erste (Filter-)String ist eine Art LIKE-Ausdruck, den Sie nicht verändern dürfen. Daher ist diese Quelltext-Zeile korrekt:

```
Dialog.Filter = ["*.png;*.jpg", ("Image files")]
```

### 5. Fall

Vorsicht ist immer geboten, wenn man u.a. Variablen - zum Beispiel sVerb="reads" - in Strings hinein konkateniert und die unterschiedliche Grammatik der Sprachen zu beachten ist:

```
Message("Do you know if she " & sVerb & " a book?") ***
```

Dieser Satz enthält übersetzbare Strings und eine (String-)Variable, deren Wert ebenso zu übersetzen ist. Er ist aber nicht gut ins Deutsche übersetzbar, weil sich die Satzstrukturen im Englischen und im Deutschen in diesem Fall unterscheiden. Während in der deutschen Sprache das Verb an das Satzenende gelegt wird: 'Weißt du, ob sie ein Buch liest?' wäre es im Englischen 'Do you know if she reads a book?'. Aus Message("Do you know if she " & sVerb & " a book?") müsste Message("Weißt du, ob sie ein Buch " & sVerb & "?") nach einer Übersetzung werden.

```
sVerb = ("reads")  
Message.Title = "TEST"  
Message(("Do you know if she") & " " & Tr(sVerb) & " " & ("a book?"))
```

Die Änderungen betreffen das Einfügen von Leerzeichen (Option) in **\*\*\***, die in den Übersetzungen nicht zu sehen wären, wenn sie am Ende stehen oder führende Leerzeichen sind und daher oft vergessen werden.

Um den Text dieser Mitteilung in der englischen Sprache zu sehen, müssen Sie in den Projekteigenschaften unter 'Umgebung' eine Umgebungsvariable mit Variable = LANG und Wert = en\_GB.utf8 temporär einfügen.

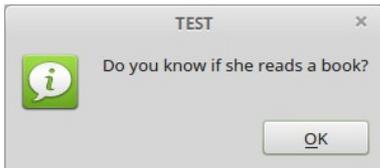


Abbildung 11.4.2.1: Original

Um die Übersetzung zu sehen, müssen Sie die Umgebungsvariable wieder löschen.

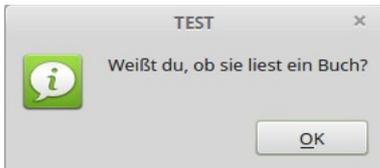


Abbildung 11.4.2.2: Übersetzung 1

Nach diesem unbefriedigenden Ergebnis für einen einfachen Satz sind Sie als Übersetzer gefragt und werden wie der Autor heftig ins Grübeln kommen! Die Lösung besteht darin, die Subst()-Funktion einzusetzen. Mit dem folgenden Quelltext-Abschnitt

```
sVerb = ("reads")
Message.Title = "TEST"
Message(Subst(("Do you know if she &1 &2?"), Tr(sVerb), ("a book")))
```

und der Übersetzung in die deutsche Sprache

```
Weißt du, ob sie &2 &1?
```

ergibt sich eine überzeugende Übersetzung:

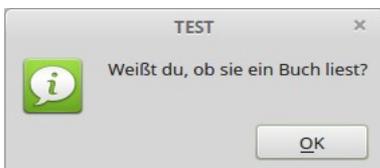


Abbildung 11.4.2.3: Übersetzung 2

Als Regel sollte gelten: Verketteten Sie übersetzbare Zeichenketten nicht mit &, sondern benutzen Sie immer die Subst()-Funktion. Die Tr()-Funktion wäre nicht notwendig, doch liest sich der Text besser, denn Tr(sVerb) bedeutet, dass die Übersetzung (Tr steht als Kürzel für Translation) des Verbs einzufügen ist. Der Einsatz von Subst() erlaubt es, die Platzhalter wie &1 &2 und Text beliebig zu verschieben und damit die Satzstruktur der Zielsprache anzupassen.

Die nächsten zwei Beispiele demonstrieren einerseits, wie die Texte strukturiert werden müssen, um eine gute Übersetzung zu präsentieren und zeigen andererseits einen anderen Ansatz, um den Aufwand bei größeren Projekten zu senken:

```
iNew = 1
bSuccess = False
Message.Title = "Information"
Message(Subst(("You have &1 new message&2!"), iNew, IIf(iNew = 1, "", ("s"))))
Message.Title = "Information"
Message(Subst(("The operation was &1successful."), IIf(bSuccess, "", ("un"))))
```

Das sind die Übersetzungen:

```
You have &1 new message&2! → Sie haben &1 neue Nachricht&2!
The operation was &1successful. → Die Operation war &1erfolgreich.
s → en
un → nicht (mit nachfolgendem Leerzeichen!)
```

Sie müssen sich schon einige Gedanken machen, um universell übersetzbaren Text mit Parametern zu schreiben oder aber zu strukturiertem, weniger natürlichen Text übergehen:

```
Message(Subst(("Number of new messages: &1"), iNew))  
Message(Subst(("Status of the operation: &1"), IIf(bSuccess, ("successful"), ("unsuccessful"))))
```

### 6. Fall

Auch die Fälle 6 und 7 verwenden die Subst()-Funktion. In den als Kommentar deklarierten ersten Zeilen finden Sie jeweils die native Form, die nicht fehlerhaft ist.

```
' txbTensileWeight.Tooltip = ("Range:") & " 10·lbs ≤ " & ("Tensile weight") & " ≤ 100·lbs" ' nativ  
txbTensileWeight.Tooltip = Subst("&1 10·lbs ≤ &2 ≤ 100·lbs", ("Range:"), ("Tensile weight"))
```

### 7. Fall

```
' Message.Warning(Str($iTensileWeight) & " " & "is an illegal draw weight value!") ' nativ  
Message.Warning(Subst("&1 &2", Str($iTensileWeight), ("is an illegal draw weight value!")))
```

### 8. Fall

Es gibt spezielle Punkte auf die Sie achten sollten, wenn eine Übersetzung in Sprachen angestrebt wird, die von rechts nach links schreiben → <http://gambaswiki.org/wiki/howto/translate>. Für diesen Fall liegen aber noch keine Erfahrungen vor.

#### 11.4.3 Internationalisierung existierender Projekte

Die Antwort auf die Frage: Kann man im Nachhinein existierende Projekte mit der Standardsprache Englisch (U.S.A.) – das ist ja der Standard bei der Anlage eines neuen Gambas-Projektes – auf eine englische Version umschreiben? ist für all jene interessant, für die erst später eine Übersetzung notwendig wird. Angenommen Sie haben ein Projekt mit allen relevanten Texten in deutscher Sprache und möchten, dass die Standardsprache Englisch bleibt, es aber eine deutsche Übersetzung geben wird. Sie verfahren zuerst wie bei einem neuen Projekt und schalten in den Optionen das Projekt auf 'übersetzbar' und wählen als Standard-Sprache Englisch mit der Lokalisation "Englisch (allgemein)". Danach müssen Sie alle deutschen Texte auf den Steuerelementen und im Quelltext manuell auf Englisch umschreiben und dann im Übersetzungsdialog jeweils die deutsche Übersetzung eintragen. Der Aufwand ist beträchtlich!

Wenn Sie darauf verzichten können, dass die Standardsprache des Projekts Englisch ist, können Sie das Projekt übersetzbar machen und als Standardsprache Deutsch eintragen. Dann legen Sie in der Projektübersetzung Englisch als neue Übersetzung an und müssen nur alle deutschen Texte auf den Steuerelementen und im Quelltext übersetzen. Dann würde aber das Projekt zum Beispiel bei einer spanischen Locale (für die Sie keine Übersetzung bereit stellen) auf Deutsch starten. Damit könnte ein spanischer Entwickler möglicherweise weniger anfangen, als wenn das Programm wie im zuvor geschilderten Fall in Englisch starten würde.

Als Empfehlung kann gelten: Entwickeln Sie Ihre Projekte möglichst mit englischen Texten auf den Steuerelementen und im Quelltext. Nur dann lassen sich jederzeit deutsche oder anderssprachige Übersetzungen leicht nachtragen. So ist gewährleistet, dass allen Nutzern in englischsprachigen Ländern oder in Ländern, für die Sie keine Übersetzung angefertigt haben, eine englische Version präsentiert wird und für alle anderen Länder eine übersetzte.

#### 11.4.4 Exportieren und importieren von Sprach-Dateien

In der Projektübersetzung gibt es den Reiter 'Importieren'. Nach dessen Aktivierung öffnet sich ein Datei-Öffnen-Dialog, in dem Sie eine Sprach-Datei \*.po für den Import in das aktuelle Projekt auswählen können. Zuvor können Sie bestimmte Präferenzen ankreuzen, sofern das notwendig wird.

Das Exportieren einer Sprach-Datei erfolgt über den Reiter 'Übersetzung' im Menü des Menü-Buttons mit dem Eintrag 'Exportieren', dem drei Untermenüs folgen. Nach der Auswahl folgt ein Datei-Speichern-Dialog für die zutreffende Sprach-Datei. Der vorgeschlagene Dateiname sollte übernommen werden und folgt der Syntax: *projektname-sprachkürzel\_der\_übersetzten\_sprache-projektversion.po* (Beispiel: spine-de-0.1.7.po).

### 11.4.5 Hinweise

Wenn Sie Tooltips für ausgewählte Steuerelemente vorgesehen haben, dann sollten diese auch übersetzt werden. Es ist von Vorteil, den Text für die Tooltips aber nicht im Objekt-Inspektor, sondern direkt im Quelltext festzulegen, weil sie so nicht aus dem Blick geraten. Im Kapitel 20.1.3 Message und I18N → <http://www.gambas-buch.de/dw/doku.php?id=k20:k20.1:k20.1.3:start> finden Sie Hinweise darauf, dass Übersetzungen in einem Projekt u.a. auch Auswirkungen auf die (notwendige) Breite von Steuerelementen haben können und deshalb unter dem Aspekt Formular-Layout zu beachten sind.

Um die IDE von Gambas für Tests temporär in der englischen Version zu starten, müssen Sie die passende Umgebungsvariable 'LANG' in einer Konsole so setzen:

```
$ LANG=en_GB.utf8 gambas3
```

### 11.4.6 Beispiel

Um die Hinweise in diesem Kapitel mindestens einmal praktisch umzusetzen, könnten Sie ein existierendes kleines Projekt (Spinel18NE) in englischer Sprache (Standard) in die deutsche Sprache übersetzen. Das Projekt-Archiv finden Sie im Download-Bereich. Die Antwort auf die Frage, warum nicht ein Projekt mit der Standardsprache Deutsch in die englische Sprache übersetzt wird – was erst einmal logischer klingt – finden Sie im ersten Teil des Kapitels und im Unter-Kapitel 11.4.3.

- Projekt so konfigurieren, dass es übersetzt werden kann.
- Standard-Sprache Englisch mit der Lokalisation "Englisch (allgemein)" auswählen.
- Die zu übersetzenden Zeichenketten auf den Steuerelementen und im Quelltext auszeichnen.
- Das Projekt (schrittweise) übersetzen.

Folgende Übersetzungen sind verwendbar:

```
"TextBox" → "-"
"Test input in a TextBox and ComboBox for given input alphabets." → "Test von Eingaben in einer TextBox und
in einer ComboBox für vorgegebene Eingabe-Aphabete."
"Spine Value Calculator for Arrows" → "Spinewert-Berechnungen für Pfeile"
"Range:" → "Wertebereich"
"Tensile weight" → "Zuggewicht"
"The value for the tensile weight is missing!" → "Es fehlt der Wert für das Zug-Gewicht!"
"Input error in the TextBox" → "Eingabe-Fehler in der Textbox"
"is an illegal draw weight value!" → "ist ein unzulässiger Zugwert!"
"Calculate" → "Berechnen"
"Reset" → "-"
"Tensile Weight:" → "Zug-Gewicht:"
"lbs" → "-"
"String class:" → "Sehnen-Klasse:"
"Number" → "Anzahl"
"With Quiver" → "Mit Köcher"
"Without Quiver" → "Ohne Köcher"
"Spine (Wood):" → "Spine (Holz)"
"Spine (Carbon):" → "-"
"(AMO)" → "-"
```

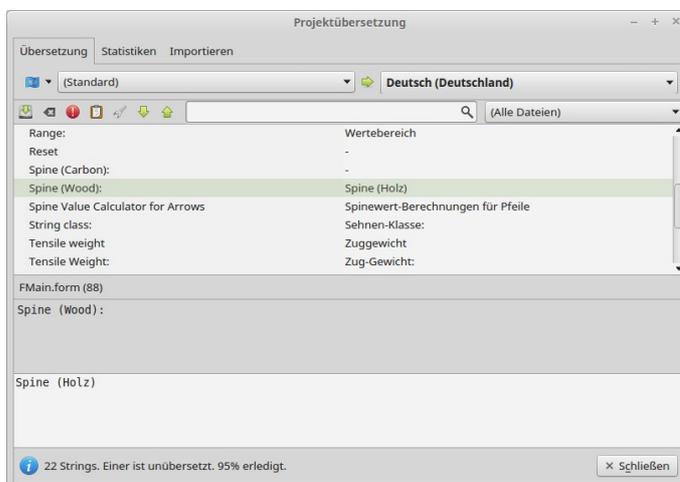


Abbildung 11.4.6.1: Projektübersetzung

Auch das Projekt *Spine118NED* mit der Übersetzung finden Sie als Projekt-Archiv im Download-Bereich.