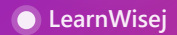


Übersetzungsworkflow: ResX Resource Manager und weitere Werkzeuge

Lektionsleitfaden · Lab- und Prüfungsleitfaden



Lektionsleitfaden

Übersetzungen als Teamarbeit behandeln: die Matrix des ResX Resource Manager über jede .resx-Datei der Projektmappe, das Hinzufügen einer Sprachspalte und das Filtern nach nicht übersetzten Einträgen, Ressourcenkommentare, die Übersetzern Kontext geben, invariante Zeichenfolgen, automatische Übersetzung nur als Entwurf, Export und Import über eine RESX-fähige Plattform und das Prüfen von Ressourcen-Diffs wie von Code. Was das für Entwicklerinnen und Entwickler bedeutet, die die Grundlagen von C# und die Wisej.NET-Controls kennen und die Anwendung nun in mehr als einer Sprache ausliefern müssen – und wie Sie dieses Modul lesen.



Fokus: Produktivbetrieb

Übersetzungen als Teamarbeit behandeln: die Matrix des ResX Resource Manager über jede .resx-Datei der Projektmappe, das Hinzufügen einer Sprachspalte und das Filtern nach nicht übersetzten Einträgen, Ressourcenkommentare, die Übersetzern Kontext geben, invariante Zeichenfolgen, automatische Übersetzung nur als Entwurf, Export und Import über eine RESX-fähige Plattform und das Prüfen von Ressourcen-Diffs wie von Code.

1. Das Problem im Produktivbetrieb

GlobalDesk trägt inzwischen eine neutrale `Resources/Strings.resx`, eine deutsche `Strings.de.resx`, Designer-Ressourcen auf dem Kundeneditor und eine `Resources-de.resx` für die Systembeschriftungen von Wisej. Italienisch mit dem Visual Studio-Editor hinzuzufügen heißt, diese Dateien Registerkarte für Registerkarte zu öffnen und Schlüssellisten nach Augenmaß zu vergleichen — und die Person, die Italienisch spricht, ist meist nicht die Person, die die Projektmappe öffnen kann. Dieses Modul macht aus der Übersetzung einen Workflow mit sichtbarer Lückenliste, einer Übergabe und einem Prüfschritt.

2. Zentrale Konzepte

Konzept	Warum es wichtig ist
Matrixsicht	Schlüssel in den Zeilen und Sprachen in den Spalten, sodass ein fehlender italienischer Wert und ein abgedrifteter deutscher Begriff auf einen Blick sichtbar sind.
ResX Resource Manager	Ein quelloffenes .NET-Lokalisierungswerkzeug, als Visual Studio-Erweiterung und als eigenständige Anwendung, das die RESX-Ressourcen einer Projektmappe in einem zentralen Raster zeigt.
Sprachspalte	Eine Zielsprache hinzuzufügen erzeugt die kulturspezifische Datei neben der neutralen und gibt Ihnen eine leere Spalte zum Füllen; die Schlüssel ändern sich nie.
Filter für nicht Übersetztes	Listet nur die Einträge ohne Wert in der gewählten Sprache auf, sodass aus „sind wir fertig?“ eine Zahl wird statt einer Meinung.
Ressourcenkommentar	Eine Notiz neben dem Wert, die einem Übersetzer sagt, ob Open ein Verb, ein Adjektiv, ein Ticketzustand oder ein Befehl ist.
Invariante Zeichenfolge	Eine Markierung auf Einträgen, die niemals übersetzt werden dürfen, etwa dem Produktnamen und technischen Bezeichnungen.
Export-Import-Umlauf	Spalten, die an einen Übersetzer oder eine RESX-fähige Plattform gehen und als kulturspezifische .resx -Dateien zurückkommen, die der Build wie jede andere Ressource behandelt.

3. Der rote Faden der Lektion

Ein Raster statt sechs Dateien

Der Ressourceneditor von Visual Studio ist für eine Datei völlig ausreichend. Er hört in dem Moment auf, gut zu sein, in dem die Frage vergleichend wird: Hat jeder Schlüssel der neutralen Datei einen deutschen Wert? Ein Übersetzungswerkzeug beantwortet das, indem es die Ressourcen als Matrix auslegt, Schlüssel in den Zeilen und jede Sprache als eigene Spalte. ResX Resource Manager ist die übliche Wahl: ein quelloffenes .NET-Werkzeug, das als Visual Studio-Erweiterung ausgeliefert wird, die die aktive Projektmappe durchsucht und ihre RESX-Ressourcen in einem zentralen Raster sammelt, und als eigenständige Anwendung.

Es passt zu Wisej.NET, weil es nichts anzupassen gibt. Wisej.NET hält lokalisierten Text in standardmäßigen .NET-RESX-Dateien, deshalb liest ein allgemeines RESX-Werkzeug GlobalDesk genauso wie jedes andere .NET-Projekt; das Wisej-Beispiel StockViewer nennt ausdrücklich die Lokalisierung mit dem Designer zusammen mit ResX Manager. Der dokumentierte Funktionsumfang ist der Workflow selbst: nebeneinanderliegende Sprachspalten, hervorgehobene nicht übersetzte Einträge, Übersetzen im Raster, Filtern und Sortieren, automatische Übersetzung, Import und Export, Erkennung von Code-Referenzen, invariante Zeichenfolgen, Snapshots, Skripting und Rechtschreibprüfung. Die Erkennung von Code-Referenzen macht sich schnell bezahlt: Zu wissen, wo ein Schlüssel verwendet wird, ist oft die einzige Möglichkeit, zu entscheiden, was eine Beschriftung aus einem Wort bedeutet.

Was das Werkzeug nicht ist: ein Teil Ihrer Anwendung. Es bearbeitet Dateien; zur Laufzeit führt GlobalDesk weiterhin eine gewöhnliche .NET-Ressourcensuche über `Wisej.Resources.ResourceManager` durch, genau wie in Modul 3. Lassen Sie ein Lokalisierungswerkzeug niemals zu einer Laufzeitabhängigkeit werden.

Italienisch hinzufügen und aus „nicht übersetzt“ eine Zahl machen

Eine Sprache hinzuzufügen erzeugt die kulturspezifische Datei neben der neutralen, `Strings.it.resx` neben `Strings.resx`, und gibt Ihnen eine neue, leere Spalte. Nichts wird neu verschlüsselt; die neutrale Ressource bleibt die einzige Wahrheit darüber, welche Schlüssel existieren. Der Filter für nicht übersetzte Einträge macht dann aus einer vagen Frage eine Liste, die Sie abarbeiten und weitergeben können.

Die sichere Reihenfolge der Arbeit gehört auf eine Karteikarte. Halten Sie `Strings.resx` vollständig und geprüft in der Versionsverwaltung, öffnen Sie den ResX Resource Manager, fügen Sie die Zielsprache hinzu, übersetzen oder importieren Sie Werte unter Beibehaltung der Schlüssel, filtern Sie nach den leeren Einträgen und räumen Sie sie ab, bauen und starten Sie unter der Zielkultur, prüfen Sie den Diff vor dem Commit, weil das Werkzeug Ihre Änderungen je nach Einstellungen sofort speichern kann, und testen Sie schließlich die Benutzeroberfläche und nicht nur das Raster.

Ein vollständiges Raster mit einer abgeschnittenen Schaltfläche ist immer noch ein Fehler, und italienische Beschriftungen sind meist länger als englische. Die gegenteilige Falle ist leiser: Ein Schlüssel ohne italienischen Wert fliegt nicht auf, er fällt auf den neutralen Wert zurück, sodass der Bildschirm fertig aussieht, während die Lücke nur im Raster lebt.

Automatische Übersetzung ist ein Entwurf, und der Diff ist die Prüfung

Automatische Übersetzung ist für eine erste Abdeckung wirklich nützlich. Sie ist ein Entwurf. Prüfen Sie Terminologie, Platzhalter, Zeichensetzung, Zugriffstasten, Produktnamen und Kontext, bevor irgendetwas davon ausgeliefert wird, denn eine Übersetzung kann technisch korrekt und in einer Benutzeroberfläche trotzdem das falsche Wort sein. Der Eintrag `Customer.LastOrder` aus Modul 3 macht das Risiko greifbar: Ein Entwurf, der seine beiden Platzhalter verliert oder vertauscht, macht aus einem Formatierungsaufruf einen Fehlschlag.

Das Mittel gegen Mehrdeutigkeit ist Kontext, und Kontext lebt im Ressourcenkommentar, der den Eintrag durch jeden Umlauf begleitet:

```
<data name="CustomerEditor.Open" xml:space="preserve">
  <value>Open</value>
  <comment>Button caption, verb: opens the selected customer.</comment>
</data>
<data name="App.ProductName" xml:space="preserve">
  <value>GlobalDesk</value>
  <comment>Product name, invariant: never translate.</comment>
</data>
```

Behandeln Sie diese Dateien wie Code: Prüfen Sie die Diffs, halten Sie Schlüssel stabil, vermeiden Sie Massenumbenennungen, während eine Runde läuft, und lassen Sie niemals einen maschinellen Entwurf eine freigegebene menschliche Übersetzung ohne Prüfung überschreiben. Weil das Werkzeug beim Tippen schreiben kann, ist der Diff in der Versionsverwaltung das Sicherheitsnetz.

Die Zeichenfolgen jemandem übergeben, der nie Visual Studio öffnet

Genau dafür gibt es Export und Import. Sie exportieren die neutrale Spalte zusammen mit der Zielspalte, der Übersetzer arbeitet in dieser Datei, und der erneute Import ordnet die zurückgelieferten Zeilen wieder ihren Schlüsseln zu, sodass Schlüssel und Platzhalter eine Person überstehen, die nie eine `.resx` gesehen hat. Was zurückkommt, muss dieselbe Schlüsselmenge sein.

Wenn die Sprachen oder die Prüfer mehr werden, zieht dieselbe Form in ein cloudbasiertes Übersetzungsmanagementsystem um. Zu den Plattformen, die `.resx` unterstützen, gehören Crowdin, Lokalise, Phrase und POEditor, und der Umlauf bleibt unverändert: die neutrale RESX hochladen oder abgleichen, in der Plattform übersetzen und prüfen, die kulturspezifischen `.resx`-Dateien exportieren, committen und dann in der echten Oberfläche bauen, starten und prüfen. Was Sie dazukaufen, ist Steuerung: Übersetzungsspeicher, Prüferrollen und zentrale Terminologie.

Zwischen Designer-Editor, ResXManager und einer Cloud-Plattform wählen

Die praktische Leitlinie ist kurz, und es geht dabei um das Team und nicht um das Werkzeug:

- Ein Entwickler, zwei Sprachen: Der RESX-Editor von Visual Studio genügt möglicherweise schon.
- Ein Entwicklerteam, mehrere Sprachen: ResX Resource Manager, ein effizienter Workflow in der IDE.
- Externe Übersetzer: Export und Import oder eine Plattform mit RESX-Unterstützung.
- Viele Produkte, Sprachen und Prüfer: Ein Cloud-TMS ist leichter zu steuern.

Was Sie auch wählen, der Vertrag des Projekts verschiebt sich nicht: kulturspezifische `.resx`-Dateien neben der neutralen Datei mit identischen Schlüsseln, gelesen über die standardmäßige Ressourcensuche. Nehmen Sie die leichteste Variante, die nicht übersetzte Einträge sichtbar und den Diff prüfbar macht, und rüsten Sie erst auf, wenn der Umlauf zum Engpass wird.



Typische Fehler

Einen maschinellen Entwurf ausliefern, weil das Raster grün geworden ist. Einen Platzhalter in einem übersetzten Wert verlieren. Schlüssel mitten in der Runde umbenennen. Das Werkzeug Änderungen speichern lassen, die Sie nie in einem Diff gelesen haben. Den Produktnamen übersetzen, weil ihn niemand als invariant markiert hat. Die Arbeit für erledigt erklären, ohne die Anwendung in der Zielkultur auszuführen.



Gewohnheit für die Lokalisierungsschichten

Entscheiden Sie, wo jede Zeichenfolge lebt, bevor Sie sie tippen. Designer-Ressourcen gehören zu einem entworfenen Bildschirm und werden beim Erstellen seiner Controls angewendet; gemeinsame Anwendungsressourcen in `.resx` tragen den Text, den Code erzeugt oder den mehrere Bildschirme wiederverwenden; `Resources-[LANG].resx` überschreibt die Systembeschriftungen von Wisej.NET. Speichern Sie niemals ein formatiertes Datum, eine formatierte Zahl oder einen formatierten Währungsbetrag: Behalten Sie den Wert und formatieren Sie ihn zum Anzeigzeitpunkt für die aktive Kultur. Und denken Sie daran, dass die Kultur pro Sitzung gilt, sodass die Sprache eines Benutzers niemals den Bildschirm eines anderen verändern darf.



Lernziele

Am Ende dieses Moduls können Sie:

- ✓ den ResX Resource Manager nutzen, um mehrere RESX-Sprachen nebeneinander zu prüfen.
- ✓ eine Sprache hinzufügen und nicht übersetzte Einträge erkennen.
- ✓ den sicheren Umgang mit automatischer Übersetzung und Versionsverwaltung erklären.
- ✓ Übersetzungsinhalte für externe Übersetzer oder ein TMS exportieren und importieren.
- ✓ Workflows nur mit Visual Studio, mit ResXManager und in der Cloud vergleichen.

Lab- und Prüfungsleitfaden

Was Sie im praktischen Lab bauen, das vorgeschlagene Vorgehen und die geforderten Ergebnisse.



Praktisches Lab

Führen Sie eine ehrliche Übersetzungsrunde an GlobalDesk durch. Öffnen Sie die Projektmappe im ResX Resource Manager, fügen Sie Italienisch als neue Sprachspalte hinzu und listen Sie mit dem Filter für nicht übersetzte Einträge jeden Schlüssel auf, der noch keinen Wert hat. Versehen Sie die drei Schlüssel, deren Bedeutung ohne Kontext mehrdeutig ist, mit Kommentaren – beginnend mit der einwortigen Beschriftung Open –, übersetzen Sie dann die italienischen Werte oder lassen Sie sie maschinell entwerfen, und markieren Sie den Produktnamen und die beiden technischen Bezeichner als invariant, damit sie niemand übersetzt. Exportieren Sie die neutrale und die italienische Spalte, importieren Sie sie wieder und prüfen Sie den entstandenen .resx-Diff, bevor Sie ihn als eine einzige überprüfbare Änderung committen. Schließen Sie mit einer einseitigen Workflow-Notiz ab, die den Visual Studio-Editor, den ResX Resource Manager und eine Cloud-Plattform für ein Team vergleicht, dessen Übersetzer keine Entwickler sind.

Ausgangspunkt

GlobalDesk nach Modul 5: `Resources/Strings.resx` und `Strings.de.resx` tragen dieselben Schlüssel, der Kundeneditor hat deutsche Designer-Ressourcen, `Resources-de.resx` überschreibt eine Systembeschriftung von Wisej, die Sprachauswahl bietet bereits Englisch, Deutsch und Arabisch an, und das Layout übersteht lange Zeichenfolgen. Italienisch gibt es nirgends, es gibt keinen Kommentar an irgendeinem Eintrag, und die Frage „wie viel fehlt noch?“ lässt sich nur beantworten, indem man zwei Dateien nebeneinander durchscrollt.

Vorgeschlagenes Vorgehen

- ✓ Öffnen Sie, bevor Sie irgendetwas hinzufügen, `Strings.resx` und `Strings.de.resx` im Visual Studio-Editor und versuchen Sie, die Schlüssel, die der deutschen Datei fehlen, mit bloßem Auge zu finden. Stoppen Sie die Zeit; diese Zahl ist das Argument für den Rest des Labs.
- ✓ Öffnen Sie die Projektmappe im ResX Resource Manager und fügen Sie Italienisch hinzu. Bestätigen Sie, dass `Strings.it.resx` neben der neutralen Datei erscheint, dass die Schlüsselliste unberührt bleibt und dass die neue Spalte leer ist. Schalten Sie den Filter für nicht übersetzte Einträge ein und notieren Sie die Anzahl.
- ✓ Kommentieren Sie, bevor Sie übersetzen. Geben Sie `CustomerEditor.Open` einen Kommentar, der sagt, ob es ein Verb oder ein Ticketzustand ist, tun Sie dann dasselbe für die beiden anderen Schlüssel, deren Bedeutung aus dem Wert allein nicht hervorgeht, und markieren Sie den Produktnamen und die beiden technischen Bezeichner als invariant.
- ✓ Füllen Sie die italienische Spalte, wenn Sie mögen zunächst mit automatischer Übersetzung für eine erste Abdeckung, und prüfen Sie dann jeden Wert: Terminologie, Zeichensetzung und vor allem die Platzhalter in `Customer.LastOrder`. Wiederholen Sie den Filter für nicht übersetzte Einträge, bis er nichts mehr liefert.
- ✓ Exportieren Sie die neutrale und die italienische Spalte, importieren Sie dieselbe Datei wieder und lesen Sie den `git diff` von `Strings.it.resx` Zeile für Zeile, bevor Sie die Runde als eine

Änderung committen. Führen Sie GlobalDesk dann unter `it-IT` aus und prüfen Sie die Bildschirme, nicht das Raster.

Geforderte Ergebnisse

- ✓ Strings.it.resx, im ResX Resource Manager erstellt, ohne einen einzigen nicht übersetzten Schlüssel
- ✓ Ressourcenkommentare zu mindestens drei mehrdeutigen Schlüsseln, damit ein Übersetzer den Kontext hat
- ✓ Invariant-Markierungen für den Produktnamen und die Bezeichner, die nicht übersetzt werden dürfen
- ✓ Export, erneuter Import und ein geprüfter .resx-Diff, als eine Änderung committet
- ✓ Workflow-Notiz, die den Visual Studio-Editor, den ResX Resource Manager und ein Cloud-TMS vergleicht

Wiederholungsfragen

- ✓ Ihre italienische Spalte ist voll und die Anwendung sieht unter `it-IT` richtig aus, aber ein Kollege behauptet, zwei Schlüssel seien nie übersetzt worden. Wie können beide Aussagen zugleich wahr sein, und welche Funktion des Werkzeugs entscheidet den Streit?
- ✓ Ein italienischer maschineller Entwurf hat `Ultimo ordine: {1} il {0}` für einen Schlüssel geliefert, dessen neutraler Wert `Last order: {0} on {1}` lautet. Was geht kaputt, wann geht es kaputt, und was hätte es vor dem Commit aufgefangen?
- ✓ Ihr Team hat fünf Sprachen, zwei davon übersetzt von einer Agentur, die nie Visual Studio geöffnet hat. Welchen der drei Workflows würden Sie einführen, und was müsste sich ändern, bevor Sie zum nächstgrößeren wechseln?



Gewohnheit für die Lokalisierungsschichten

Entscheiden Sie, wo jede Zeichenfolge lebt, bevor Sie sie tippen. Designer-Ressourcen gehören zu einem entworfenen Bildschirm und werden beim Erstellen seiner Controls angewendet; gemeinsame Anwendungsressourcen in .resx tragen den Text, den Code erzeugt oder den mehrere Bildschirme wiederverwenden; Resources-[LANG].resx überschreibt die Systembeschriftungen von Wisej.NET. Speichern Sie niemals ein formatiertes Datum, eine formatierte Zahl oder einen formatierten Währungsbetrag: Behalten Sie den Wert und formatieren Sie ihn zum Anzeigzeitpunkt für die aktive Kultur. Und denken Sie daran, dass die Kultur pro Sitzung gilt, sodass die Sprache eines Benutzers niemals den Bildschirm eines anderen verändern darf.