

Flusso di lavoro della traduzione: ResX Resource Manager e altri strumenti

Guida della lezione · Guida al lab / esame



Guida della lezione

Trattare le traduzioni come lavoro di squadra: la matrice di ResX Resource Manager su tutti i .resx della soluzione, l'aggiunta di una colonna lingua e il filtro delle voci non tradotte, i commenti delle risorse che danno contesto ai traduttori, le stringhe invarianti, la traduzione automatica solo come bozza, esportazione e importazione con una piattaforma che gestisce il formato RESX, e la revisione dei diff delle risorse come si fa con il codice. Che cosa significa per uno sviluppatore che conosce le basi di C# e i controlli Wisej.NET e deve ora rilasciare l'applicazione in più di una lingua — e come affrontare questo modulo.



Focus di produzione

Trattare le traduzioni come lavoro di squadra: la matrice di ResX Resource Manager su tutti i .resx della soluzione, l'aggiunta di una colonna lingua e il filtro delle voci non tradotte, i commenti delle risorse che danno contesto ai traduttori, le stringhe invarianti, la traduzione automatica solo come bozza, esportazione e importazione con una piattaforma che gestisce il formato RESX, e la revisione dei diff delle risorse come si fa con il codice.

1. Il problema in produzione

GlobalDesk porta ormai un `Resources/Strings.resx` neutro, uno `Strings.de.resx` tedesco, le risorse del designer sull'editor cliente e un `Resources-de.resx` per le etichette di sistema di Wisej. Aggiungere l'italiano con l'editor di Visual Studio significa aprire quei file una scheda per volta e confrontare a occhio gli elenchi delle chiavi, e la persona che parla italiano di solito non è la persona che sa aprire la soluzione. Questo modulo trasforma la traduzione in un flusso di lavoro con un elenco visibile delle lacune, una consegna e un passaggio di revisione.

2. Concetti chiave

Concetto	Perché è importante
Vista a matrice	Le chiavi sulle righe e le lingue sulle colonne, così un valore italiano mancante e un termine tedesco che è andato alla deriva si vedono entrambi in un colpo solo.
ResX Resource Manager	Uno strumento di localizzazione .NET open source, estensione di Visual Studio e applicazione standalone, che mostra le risorse RESX di una soluzione in un'unica griglia centrale.
Colonna lingua	Aggiungere una lingua di destinazione crea il file specifico per cultura accanto a quello neutro e ti dà una colonna vuota da riempire; le chiavi non cambiano mai.
Filtro delle voci non tradotte	Elenca solo le voci senza valore nella lingua selezionata, così "abbiamo finito?" diventa un numero invece che un'opinione.
Commento della risorsa	Una nota conservata accanto al valore che dice a un traduttore se Open è un verbo, un aggettivo, uno stato del ticket o un comando.
Stringa invariante	Un marcatore sulle voci che non devono mai essere tradotte, come il nome del prodotto e gli identificatori tecnici.
Giro di esportazione / importazione	Colonne inviate a un traduttore o a una piattaforma che gestisce il formato RESX e restituite come file .resx specifici per cultura che la build tratta come qualsiasi altra risorsa.

3. Il filo della lezione

Una griglia invece di sei file

L'editor di risorse di Visual Studio va benissimo per un file solo. Smette di andare bene nel momento in cui la domanda diventa comparativa: ogni chiave del file neutro ha un valore tedesco? Uno strumento di traduzione risponde disponendo le risorse come una matrice, le chiavi sulle righe e ogni lingua come colonna a sé. ResX Resource Manager è la scelta abituale: uno strumento .NET open source distribuito come estensione di Visual Studio, che scandisce la soluzione attiva e raccoglie le sue risorse RESX in un'unica griglia centrale, e come applicazione standalone.

Si adatta a Wisej.NET perché non c'è nulla da adattare. Wisej.NET tiene il testo localizzato in file RESX .NET standard, quindi uno strumento RESX generico legge GlobalDesk allo stesso modo in cui legge qualsiasi altro progetto .NET; l'esempio StockViewer di Wisej segnala la localizzazione fatta con il designer insieme a ResX Manager. L'insieme delle funzionalità documentate è il flusso di lavoro stesso: colonne di lingua affiancate, voci non tradotte evidenziate, traduzione dentro la griglia, filtri e ordinamenti, traduzione automatica, importazione ed esportazione, rilevamento dei riferimenti nel codice, stringhe invarianti, snapshot, scripting e controllo ortografico. Il rilevamento dei riferimenti nel codice si guadagna il posto in fretta: sapere dove viene usata una chiave è spesso l'unico modo per decidere che cosa significa un'etichetta di una sola parola.

Quello che lo strumento non è, è parte della tua applicazione. Modifica file; a runtime GlobalDesk esegue ancora la normale ricerca di risorse .NET attraverso `Wisej.Resources.ResourceManager`, esattamente come nel Modulo 3. Non lasciare mai che uno strumento di localizzazione diventi una dipendenza a runtime.

Aggiungere l'italiano e trasformare "non tradotto" in un numero

Aggiungere una lingua crea il file specifico per cultura accanto a quello neutro, `Strings.it.resx` accanto a `Strings.resx`, e ti dà una colonna nuova e vuota. Nulla viene ri-chiavato; la risorsa neutra resta l'unica fonte di verità su quali chiavi esistono. Il filtro delle voci non tradotte trasforma poi una domanda vaga in un elenco che puoi lavorare e consegnare.

Vale la pena scrivere su un cartoncino l'ordine di lavoro sicuro. Tieni `Strings.resx` completo e rivisto nel controllo di versione, apri ResX Resource Manager, aggiungi la lingua di destinazione, traduci o importa i valori preservando le chiavi, filtra le voci vuote e svuotalo, compila ed esegui sotto la cultura di destinazione, rivedi il diff prima di committare perché a seconda delle sue impostazioni lo strumento può salvare le tue modifiche subito, e infine testa l'interfaccia utente e non solo la griglia.

Una griglia piena con un pulsante troncato resta un bug, e le etichette italiane sono tipicamente più lunghe di quelle inglesi. La trappola opposta è più silenziosa: una chiave senza valore italiano non esplode, ricade sul valore neutro, così la schermata sembra finita mentre la lacuna vive solo nella griglia.

La traduzione automatica è una bozza, e il diff è la revisione

La traduzione automatica è davvero utile per una prima copertura. È una bozza. Rivedi terminologia, segnaposto, punteggiatura, acceleratori, nomi di prodotto e contesto prima che qualsiasi cosa venga rilasciata, perché una traduzione può essere tecnicamente corretta e restare comunque la scelta di parole sbagliata in un'interfaccia utente. La voce `Customer.LastOrder` del Modulo 3 rende concreto il rischio: una bozza che perde o riordina i suoi due segnaposto trasforma una chiamata di formattazione in un errore.

Il rimedio all'ambiguità è il contesto, e il contesto vive nel commento della risorsa, che viaggia con la voce attraverso ogni giro di andata e ritorno:

```
<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>
```

Tratta quei file come codice: rivedi i diff, tieni stabili le chiavi, evita rinomine di massa mentre un giro è in corso, e non lasciare mai che una bozza automatica sovrascriva senza revisione una traduzione umana approvata. Poiché lo strumento può scrivere mentre digiti, il diff del controllo di versione è la rete di sicurezza.

Consegnare le stringhe a chi non apre mai Visual Studio

Esportazione e importazione esistono esattamente per questo. Esporti la colonna neutra insieme alla colonna di destinazione, il traduttore lavora in quel file, e la reimportazione riaggancia alle loro chiavi le righe restituite, così chiavi e segnaposto sopravvivono a una persona che non ha mai visto un `.resx`. Quello che torna indietro deve essere lo stesso insieme di chiavi.

Quando le lingue o i revisori si moltiplicano, la stessa forma si sposta su un sistema cloud di gestione della traduzione. Tra le piattaforme che supportano `.resx` ci sono Crowdin, Lokalise, Phrase e POEditor, e il giro di andata e ritorno non cambia: carichi o sincronizzi il RESX neutro, traduci e rivedi nella piattaforma, esporti i file `.resx` specifici per cultura, li committi, poi compili, esegui e verifichi nell'interfaccia reale. Quello che compri è la governance: memoria di traduzione, ruoli di revisione e terminologia centralizzata.

Scegliere tra l'editor del designer, ResXManager e una piattaforma cloud

La guida pratica è breve, e riguarda il team più che lo strumento:

- Un solo sviluppatore, due lingue: l'editor RESX di Visual Studio può già bastare.
- Un team di sviluppatori, più lingue: ResX Resource Manager, un flusso di lavoro efficiente dentro l'IDE.
- Traduttori esterni: esportazione e importazione, oppure una piattaforma con supporto RESX.
- Molti prodotti, lingue e revisori: un TMS cloud è più facile da governare.

Qualunque cosa tu scelga, il contratto del progetto non si muove: file `.resx` specifici per cultura accanto al file neutro con chiavi identiche, letti attraverso la normale ricerca delle risorse. Prendi l'opzione più leggera che rende visibili le voci non tradotte e rivedibile il diff, e passa a quella successiva solo quando il giro di andata e ritorno diventa il collo di bottiglia.



Errori tipici

Rilasciare una bozza automatica perché la griglia è diventata verde. Perdere un segnaposto in un valore tradotto. Rinominare le chiavi a metà del giro. Lasciare che lo strumento salvi modifiche che non hai mai letto in un diff. Tradurre il nome del prodotto perché nessuno l'ha marcato come invariante. Dichiarare il lavoro finito senza eseguire l'applicazione nella cultura di destinazione.



La regola dei livelli di localizzazione

Decidi dove vive ogni stringa prima di scriverla. Le risorse del designer appartengono a una singola schermata disegnata e vengono applicate quando i suoi controlli vengono creati; le risorse condivise dell'applicazione nei file `.resx` portano il testo che il codice genera o che più schermate riutilizzano; `Resources-[LANG].resx` sovrascrive le etichette di sistema di Wisej.NET. Non memorizzare mai una data, un numero o un importo già formattati: conserva il valore e formattalo per la cultura attiva al momento di mostrarlo. E ricorda che la cultura è per sessione, quindi la lingua di un utente non deve mai cambiare la schermata di un altro.



Obiettivi di apprendimento

Alla fine di questo modulo sarai in grado di:

- ✓ Usare ResX Resource Manager per rivedere più lingue RESX affiancate.
- ✓ Aggiungere una lingua e individuare le voci non tradotte.
- ✓ Spiegare l'uso sicuro della traduzione automatica e del controllo di versione.
- ✓ Esportare/importare il contenuto della traduzione per traduttori esterni o un TMS.
- ✓ Confrontare i flussi di lavoro di localizzazione con il solo Visual Studio, con ResXManager e con il cloud.

Guida al lab / esame

Che cosa costruirai nel lab pratico, l'approccio suggerito e i deliverable richiesti.



Lab pratico

Esegui un giro di traduzione onesto su GlobalDesk. Apri la soluzione in ResX Resource Manager, aggiungi l'italiano come nuova colonna lingua e usa il filtro delle voci non tradotte per elencare ogni chiave ancora senza valore. Aggiungi commenti alle tre chiavi il cui significato è ambiguo fuori contesto, a partire dall'etichetta di una sola parola Open, poi traduci o fai tradurre in bozza dalla macchina i valori italiani e marca come invarianti il nome del prodotto e i due identificatori tecnici, così che nessuno li traduca. Esporta la colonna neutra e quella italiana, reimportale e rivedi il diff del .resx risultante prima di committarlo come un'unica modifica rivedibile. Concludi con una nota di una pagina sul flusso di lavoro che confronta l'editor di Visual Studio, ResX Resource Manager e una piattaforma cloud per un team i cui traduttori non sono sviluppatori.

Punto di partenza

GlobalDesk dopo il Modulo 5: `Resources/Strings.resx` e `Strings.de.resx` portano le stesse chiavi, l'editor cliente ha le risorse tedesche del designer, `Resources-de.resx` sovrascrive un'etichetta di sistema di Wisej, il selettore della lingua offre già l'inglese, il tedesco e l'arabo, e il layout sopravvive alle stringhe lunghe. Non c'è italiano da nessuna parte, non c'è un commento su una sola voce, e non c'è modo di rispondere a "quanto manca?" se non scorrendo due file affiancati.

Approccio suggerito

- ✓ Prima di aggiungere qualsiasi cosa, apri `Strings.resx` e `Strings.de.resx` nell'editor di Visual Studio e prova a trovare a occhio le chiavi che mancano al file tedesco. Cronometrati; quel numero è l'argomento a favore del resto del lab.
- ✓ Apri la soluzione in ResX Resource Manager e aggiungi l'italiano. Verifica che `Strings.it.resx` compaia accanto al file neutro, che l'elenco delle chiavi sia intatto e che la nuova colonna sia vuota. Attiva il filtro delle voci non tradotte e annota il conteggio.
- ✓ Commenta prima di tradurre. Dai a `CustomerEditor.Open` un commento che dica se è un verbo o uno stato del ticket, poi fai lo stesso per le altre due chiavi il cui significato non è ovvio dal solo valore, e marca come invarianti il nome del prodotto e i due identificatori tecnici.
- ✓ Riempi la colonna italiana, eventualmente usando la traduzione automatica per una prima copertura, poi rivedi ogni valore: terminologia, punteggiatura e soprattutto i segnaposto in `Customer.LastOrder`. Riesegui il filtro delle voci non tradotte finché non restituisce nulla.
- ✓ Esporta la colonna neutra e quella italiana, reimporta lo stesso file, e leggi riga per riga il `git diff` di `Strings.it.resx` prima di committare il giro come un'unica modifica. Poi esegui GlobalDesk sotto `it-IT` e controlla le schermate, non la griglia.

Deliverable richiesti

- ✓ `Strings.it.resx` prodotto con ResX Resource Manager senza nessuna chiave non tradotta
- ✓ Commenti di risorsa su almeno tre chiavi ambigue perché un traduttore abbia il contesto

- ✓ Marcatori di invarianza sul nome del prodotto e sugli identificatori che non devono essere tradotti
- ✓ Esportazione, reimportazione e un diff .resx rivisto e committato come un'unica modifica
- ✓ Nota sul flusso di lavoro che confronta l'editor di Visual Studio, ResX Resource Manager e un TMS cloud

Domande di autovalutazione

- ✓ La tua colonna italiana è piena e l'applicazione sembra corretta in `it-IT`, ma un collega sostiene che due chiavi non sono mai state tradotte. Come possono essere vere entrambe le affermazioni allo stesso tempo, e quale funzionalità dello strumento chiude la discussione?
- ✓ Una bozza automatica ha restituito `Ultimo ordine: {1} il {0}` per una chiave il cui valore neutro è `Last order: {0} on {1}`. Che cosa si rompe, quando si rompe, e che cosa l'avrebbe intercettato prima del commit?
- ✓ Il tuo team ha cinque lingue, due delle quali tradotte da un'agenzia che non ha mai aperto Visual Studio. Quale dei tre flussi di lavoro adoteresti, e che cosa dovrebbe cambiare prima di passare a quello successivo?



La regola dei livelli di localizzazione

Decidi dove vive ogni stringa prima di scriverla. Le risorse del designer appartengono a una singola schermata disegnata e vengono applicate quando i suoi controlli vengono creati; le risorse condivise dell'applicazione nei file .resx portano il testo che il codice genera o che più schermate riutilizzano; Resources-[LANG].resx sovrascrive le etichette di sistema di Wisej.NET. Non memorizzare mai una data, un numero o un importo già formattati: conserva il valore e formattalo per la cultura attiva al momento di mostrarlo. E ricorda che la cultura è per sessione, quindi la lingua di un utente non deve mai cambiare la schermata di un altro.