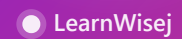


Localizzare l'interfaccia Wisej.NET nel designer di Visual Studio

Guida della lezione · Guida al lab / esame



Guida della lezione

La localizzazione nel designer su un contenitore Wisej.NET di primo livello: la proprietà Localizable, il selettore Language del designer, le proprietà che possono legittimamente cambiare da una lingua all'altra come Text, Size, Location, colori e immagini, i file di risorse che il designer genera, e la regola per cui le risorse del designer vengono applicate quando i controlli vengono creati. 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

La localizzazione nel designer su un contenitore Wisej.NET di primo livello: la proprietà Localizable, il selettore Language del designer, le proprietà che possono legittimamente cambiare da una lingua all'altra come Text, Size, Location, colori e immagini, i file di risorse che il designer genera, e la regola per cui le risorse del designer vengono applicate quando i controlli vengono creati.

1. Il problema in produzione

Il Modulo 1 ha dato a GlobalDesk una dashboard le cui etichette arrivano tutte da `Strings.resx`, ed è quella la casa giusta per il testo che il codice genera. L'editor cliente è diverso: è una schermata disegnata, sistemata a occhio nel designer di Visual Studio. Far passare ognuna delle sue etichette da una chiamata a un helper significa modificare a mano il file del designer, e non risolve comunque il problema vero, cioè che un *Kundennummer speichern* tedesco non ci sta dentro un pulsante dimensionato per *Save*. Questo modulo usa il flusso di localizzazione che Visual Studio offre già, così che le etichette e la geometria di una schermata disegnata possano cambiare da una lingua all'altra senza stringhe scritte nel codice e senza un secondo file di layout.

2. Concetti chiave

Concetto	Perché è importante
<code>Localizable</code>	Una proprietà a design time su un contenitore di primo livello; <code>true</code> porta il designer nel flusso di localizzazione, così i valori possono essere memorizzati per cultura.
Selettore <code>Language</code> del designer	Compare appena <code>Localizable</code> è <code>true</code> ; la cultura selezionata è la versione che stai modificando, quindi ogni cambiamento viene registrato solo per quella cultura.
La lingua neutra	La versione predefinita su cui ricade ogni cultura, e l'unica in cui aggiungi, rimuovi o ristrutturati i controlli; le viste di cultura contengono sovrascritture, non una seconda schermata.
Proprietà localizzabili	Non solo <code>Text</code> : <code>Size</code> , <code>Location</code> , colori e immagini possono avere ciascuno un valore diverso per cultura, perché la traduzione cambia la geometria oltre alle parole.
File di risorse generati	Il designer scrive i valori localizzati in file di risorse invece che nel codice C# del designer, tenendo i valori specifici di una lingua fuori dal codice che leggi e fondi ogni giorno.
Applicati al momento della creazione	I valori localizzati nel designer vengono applicati mentre i controlli vengono creati, quindi cambiare più tardi la cultura della sessione non riscrive un contenitore che esiste già.
Layout tollerante alla traduzione	Docking, ancoraggio, <code>FlowLayoutPanel</code> , <code>TableLayoutPanel</code> , <code>FlexLayoutPanel</code> , <code>AutoSize</code> e padding lasciano crescere il testo da solo, così servono meno sovrascritture di geometria.

3. Il filo della lezione

Attivare Localizable

La localizzazione nel designer si attiva sul contenitore, non sui singoli controlli. Apri in modalità di progettazione un `Form` , una `Page` , uno `UserControl` o un `Desktop` di Wisej.NET, seleziona il contenitore stesso nella finestra Properties e imposta `Localizable` a `true` . Da quel momento ogni controllo che il contenitore possiede è coperto dal flusso; la proprietà non si imposta mai su un `Button` o una `Label` figli.

Fallo dopo che il layout neutro è finito, non prima. Nel lab completi prima lo user control `CustomerEditor` inglese di GlobalDesk: campi, group box, pulsanti Save e Cancel, docking. Attivare l'interruttore a metà strada significa prendere decisioni strutturali mentre è selezionata una cultura, così una decisione che appartiene a tutte le lingue viene registrata per una sola di esse.

Modificare attraverso il selettore Language

Con `Localizable` attivo, il designer espone sul contenitore una selezione `Language` . Lasciala sulla voce predefinita per modificare la versione neutra. Scegli una cultura come il tedesco e la superficie di progettazione diventa quella versione localizzata della schermata: mostra ancora il tuo layout, ma ogni proprietà che cambi da quel momento viene registrata come sovrascrittura per quella sola cultura.

La disciplina è semplice: cambia solo ciò che deve essere diverso. Nel lab passi al tedesco, traduci le etichette, allarghi i due pulsanti più lunghi e sposti una label, e non tocchi nient'altro. Poi torni alla lingua predefinita prima di qualsiasi modifica strutturale, perché aggiungere un controllo mentre è selezionato il tedesco è il modo in cui un team si ritrova con una schermata che ha un controllo in una lingua e non in un'altra.

Più di Text: Size, Location, colori e immagini

Wisej.NET indica `Size`, `Location`, `Text`, colori e immagini come proprietà che possono cambiare da una lingua all'altra, e quell'elenco non è decorativo. I nomi composti tedeschi sono lunghi, quindi un pulsante dimensionato per *Save* tronca *Speichern*. Le finestre di dialogo francesi hanno spesso bisogno di più spazio verticale. Un'illustrazione che si legge chiaramente in un mercato può non voler dire nulla in un altro, quindi anche un' `Image` è un valore localizzato legittimo.

I caratteri mnemonici rientrano nella stessa categoria: è l'etichetta tradotta a decidere quale lettera acceleratrice è disponibile e sensata in quella lingua, quindi non dare mai per scontato che quella inglese si possa riportare tale e quale. E anche se i colori sono localizzabili, cambiarli per cultura è più raro che cambiare la geometria: tratta un colore per lingua come una decisione che sai spiegare, non come un riflesso.

Che cosa scrive il designer, e dove finisce

Quando modifichi la vista di una cultura, il designer non aggiunge un ramo a `InitializeComponent`. Memorizza i valori localizzati delle proprietà in file di risorse generati accanto al contenitore, un set per ogni cultura che hai modificato, tenendo i valori specifici di una lingua fuori dal codice C# principale del designer. È questa separazione a rendere il flusso sostenibile: il tuo codice del designer resta leggibile, i valori per lingua sono dati, e chi rivede può leggere un file di risorse senza leggere C#.

Traccia anche un confine per il resto dell'applicazione. Le risorse del designer appartengono a una sola schermata disegnata, quindi etichette fisse, testo dei pulsanti, tooltip, immagini statiche e geometria per lingua stanno bene lì. Tutto ciò che viene generato nel codice, ogni messaggio prodotto da un servizio, ogni testo di validazione e tutto ciò che viene riutilizzato da più di una schermata appartiene a un file di risorse dell'applicazione come `Strings.resx` letto tramite `Wisej.Resources.ResourceManager` — esattamente ciò che ha costruito il Modulo 1. GlobalDesk tiene entrambi.

Le risorse del designer vengono applicate quando i controlli vengono creati

È la regola che sorprende. I valori localizzati nel designer vengono applicati mentre l'albero dei controlli viene costruito. Impostare `Application.CurrentCulture` dopo cambia la cultura della sessione, e il codice che legge le risorse da quel momento in poi prende la nuova lingua, ma non torna indietro a riscrivere le proprietà di controlli che esistono già. Un utente che cambia lingua e vede metà editor ancora in inglese ha incontrato questa regola, non un bug.

Hai due possibilità oneste: ricreare il contenitore localizzato così che i suoi valori di risorsa vengano applicati sotto la nuova cultura, oppure tenerlo e scrivere del codice che riapplica tu stesso le proprietà che vengono dalle risorse. Per una schermata di queste dimensioni ricrearla è più semplice, ed è per questo che il lab aggiunge un pulsante *Recreate editor* che distrugge il controllo e ne costruisce uno nuovo.

```
private void btnRecreateEditor_Click(object sender, EventArgs e)
{
    // Le risorse del designer vengono applicate mentre i controlli vengono creati,
    // quindi la nuova cultura raggiunge questa schermata solo con una ricostruzione.
    this.hostPanel.Controls.Remove(this.customerEditor);
    this.customerEditor.Dispose();

    this.customerEditor = new CustomerEditor { Dock = DockStyle.Fill };
    this.hostPanel.Controls.Add(this.customerEditor);
}
```

Progettare layout che sopravvivono a traduzioni più lunghe

Ogni sovrascrittura di geometria che registri è un valore che qualcuno dovrà mantenere in ogni lingua futura. La sovrascrittura più economica è quella di cui non hai mai avuto bisogno, quindi costruisci il layout neutro con cose che si adattano: metti un pannello in dock invece di fissarlo, ancora un pulsante al bordo destro invece di bloccarne la `Location`, lascia che un `FlowLayoutPanel`, un `TableLayoutPanel` o un `FlexLayoutPanel` disponga una riga, attiva `AutoSize` dove è l'etichetta a dover decidere la propria larghezza, e lascia del padding perché una parola più lunga abbia dove andare.

Poi dimostrarlo. La vista del designer non è la risposta definitiva: esegui GlobalDesk nel browser sotto la cultura di destinazione. Controlla etichette troncate, pulsanti che si sovrappongono, label che vanno a capo dentro quella vicina, un ordine di tabulazione che non segue più l'ordine visivo, le immagini e ogni finestra di dialogo che la schermata può aprire. Il lab si chiude con una nota che elenca quali proprietà hai sovrascritto per il tedesco e quali hai deliberatamente lasciato al layout neutro, e il secondo elenco è quello interessante: è la prova che è il tuo layout a fare il lavoro, e non le tue sovrascritture.



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:

- ✓ Attivare la localizzazione su un Form, una Page, uno UserControl o un Desktop di Wisej.NET.
- ✓ Usare il selettore Language del designer per modificare le proprietà localizzate.
- ✓ Spiegare quali proprietà cambiano comunemente da una lingua all'altra: Text, Size, Location, Color e Images.
- ✓ Capire che cosa scrive il designer nei file .resx.

- ✓ Progettare layout che sopravvivono a traduzioni più lunghe.

Guida al lab / esame

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



Lab pratico

Localizza l'editor cliente di GlobalDesk tramite il designer di Visual Studio. Completa prima il layout neutro in inglese dello user control `CustomerEditor`, poi imposta `Localizable` a `true` e usa il selettore `Language` del designer per aggiungere una variante tedesca, cambiando solo ciò che deve essere diverso: le etichette, la larghezza dei due pulsanti più lunghi e la `Location` di una label. Esegui l'applicazione sotto `de-DE` e confronta il risultato reale nel browser con la vista del designer, poi aggiungi un pulsante `Recreate editor` che distrugge e ricostruisce il controllo, così da vedere con i tuoi occhi perché le risorse del designer non seguono un cambio di cultura a runtime. Concludi con una nota di lab che elenca le proprietà che hai sovrascritto e quelle che hai deliberatamente lasciato al layout neutro.

Punto di partenza

Apri GlobalDesk dal Modulo 1, dove la pagina dashboard legge già ogni etichetta da `Resources/Strings.resx` tramite l'helper `Texts` e formatta i valori della sua anteprima con `Application.CurrentCulture`. In questo lab lavori sullo user control `CustomerEditor` che il pulsante `Customers` della dashboard apre: i campi del codice cliente, del nome, della città e delle note con le loro label, un group box e i pulsanti `Save` e `Cancel`.

Approccio suggerito

- ✓ Completa il layout neutro in inglese di `CustomerEditor` prima di localizzare qualsiasi cosa: metti in dock il campo delle note, ancora `Save` e `Cancel` in basso a destra, colloca le coppie label/editor in un contenitore di layout come un `TableLayoutPanel` o un `FlexLayoutPanel`, e imposta `AutoSize` sulle label che devono dimensionarsi da sole.
- ✓ Seleziona nel designer lo user control stesso, imposta `Localizable` a `true` nella finestra `Properties` e verifica che ora compaia la selezione `Language`; lasciala sulla voce predefinita e ricompila una volta, così che la versione neutra sia la base.
- ✓ Porta la selezione `Language` sul tedesco e traduci solo le etichette — le label, l'intestazione del group box, `Save` e `Cancel`. Poi allarga i due pulsanti il cui testo tedesco non ci sta più cambiandone la `Size`, e sposta una label con la sua `Location` così che la colonna resti allineata. Torna alla lingua predefinita prima di toccare qualsiasi cosa strutturale.
- ✓ Esegui GlobalDesk sotto `de-DE` e controlla il browser, non il designer: cerca etichette troncate, controlli sovrapposti, una label che va a capo dentro quella vicina, l'ordine di tabulazione e le finestre di dialogo che l'editor può aprire. Annota ogni differenza che trovi tra la vista del designer e la pagina reale.
- ✓ Aggiungi alla pagina che lo ospita un pulsante **Recreate editor** che rimuove dal contenitore il `CustomerEditor` attuale, lo distrugge e ne aggiunge uno costruito da zero. Cambia la cultura, premi il pulsante e guarda le etichette finalmente seguire; poi scrivi la nota che elenca ogni proprietà che hai sovrascritto per il tedesco e ognuna che hai lasciato al layout neutro.

Deliverable richiesti

- ✓ User control CustomerEditor con Localizable impostato a true sopra un layout neutro inglese completo
- ✓ Variante di lingua tedesca nel designer che sovrascrive le etichette più almeno una Size e una Location
- ✓ Ancoraggio, docking o AutoSize applicati in modo che le etichette tedesche non possano troncarsi
- ✓ Pulsante Recreate editor che distrugge e ricostruisce il contenitore dopo un cambio di cultura
- ✓ Nota di lab che elenca le proprietà sovrascritte e quelle lasciate al layout neutro

Domande di autovalutazione

- ✓ Un collega aggiunge un nuovo campo telefono a `CustomerEditor` mentre nell'elenco `Language` è selezionato il tedesco. Che cosa vedrà probabilmente l'utente inglese, e qual è la regola che l'avrebbe evitato?
- ✓ La conferma di salvataggio che l'editor mostra dopo un aggiornamento riuscito continua ad arrivare da `Strings.resx` invece che dalle risorse del designer. Dai due ragioni per cui è quello il posto giusto, e indica un'etichetta di questa schermata che invece appartiene davvero al designer.
- ✓ Il tuo pulsante Recreate editor funziona, ma un collega sostiene che la schermata dovrebbe semplicemente riapplicare le proprietà localizzate invece di ricostruirsi. Che cosa dovrebbe sapere quel codice che la ricostruzione non ha bisogno di sapere, e quando il suo approccio sarebbe il compromesso migliore?



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.