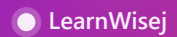


Localizar la interfaz de Wisej.NET en el diseñador de Visual Studio

Guía de la lección · Guía del laboratorio / examen



Guía de la lección

La localización en el diseñador sobre un contenedor de nivel superior de Wisej.NET: la propiedad Localizable, el selector Language del diseñador, las propiedades que legítimamente difieren según el idioma como Text, Size, Location, colores e imágenes, los archivos de recursos que genera el diseñador y la regla de que los recursos del diseñador se aplican al crearse los controles. Qué significa esto para un desarrollador que conoce los fundamentos de C# y los controles de Wisej.NET y ahora tiene que publicar la aplicación en más de un idioma, y cómo leer este módulo.



Enfoque de producción

La localización en el diseñador sobre un contenedor de nivel superior de Wisej.NET: la propiedad Localizable, el selector Language del diseñador, las propiedades que legítimamente difieren según el idioma como Text, Size, Location, colores e imágenes, los archivos de recursos que genera el diseñador y la regla de que los recursos del diseñador se aplican al crearse los controles.

1. El problema en producción

El Módulo 1 le dio a GlobalDesk un dashboard cuyas etiquetas proceden todas de `Strings.resx`, y ese es el sitio adecuado para el texto que genera el código. El editor de clientes es distinto: es una pantalla diseñada, colocada a ojo en el diseñador de Visual Studio. Hacer pasar cada una de sus etiquetas por una llamada al helper obliga a editar a mano el archivo del diseñador, y sigue sin resolver el problema real: un *Kundennummer speichern* alemán no cabe en un botón dimensionado para *Save*. Este módulo usa el flujo de localización que Visual Studio ya ofrece, de modo que las etiquetas y la geometría de una pantalla diseñada puedan diferir por idioma sin cadenas escritas en el código y sin un segundo archivo de diseño.

2. Conceptos clave

Concepto	Por qué importa
<code>Localizable</code>	Una propiedad de tiempo de diseño en un contenedor de nivel superior; <code>true</code> hace que el diseñador entre en el flujo de localización para que los valores se puedan guardar por cultura.
Selector <code>Language</code> del diseñador	Aparece en cuanto <code>Localizable</code> está en <code>true</code> ; la cultura seleccionada es la versión que estás editando, así que cada cambio se registra solo contra esa cultura.
El idioma neutro	La versión predeterminada a la que se repliegan todas las culturas, y la única en la que añades, quitas o reestructuras controles; las vistas de cultura guardan sobrescrituras, no una segunda pantalla.
Propiedades localizables	No solo <code>Text</code> : <code>Size</code> , <code>Location</code> , los colores y las imágenes pueden llevar un valor distinto por cultura, porque la traducción cambia la geometría además de las palabras.
Archivos de recursos generados	El diseñador escribe los valores localizados en archivos de recursos en lugar de en el código C# del diseñador, lo que mantiene los valores específicos de cada idioma fuera del código que lees y fusionas a diario.
Aplicados al crearse	Los valores localizados en el diseñador se aplican mientras se crean los controles, así que cambiar la cultura de la sesión más tarde no reescribe un contenedor que ya existe.
Diseño tolerante a las traducciones	El docking, el anchoring, <code>FlowLayoutPanel</code> , <code>TableLayoutPanel</code> , <code>FlexLayoutPanel</code> , <code>AutoSize</code> y el padding dejan que el texto crezca por sí solo, así que necesitas menos sobrescrituras de geometría.

3. Desarrollo de la lección

Activar Localizable

La localización en el diseñador se activa en el contenedor, no en los controles individuales. Abre un `Form` , una `Page` , un `UserControl` o un `Desktop` de Wisej.NET en modo de diseño, selecciona el propio contenedor en la ventana Properties y pon `Localizable` en `true` . A partir de ahí, todos los controles que contiene quedan cubiertos por el flujo; nunca estableces la propiedad en un `Button` o un `Label` hijo.

Hazlo después de terminar el diseño neutro, no antes. En el laboratorio completas primero el control de usuario `CustomerEditor` de GlobalDesk en inglés: campos, group box, botones Save y Cancel, docking. Activar el interruptor a medias significa tomar decisiones estructurales con una cultura seleccionada, así que una decisión que pertenece a todos los idiomas queda registrada contra uno solo.

Editar con el selector Language

Con `Localizable` activado, el diseñador muestra una selección de `Language` en el contenedor. Déjala en la entrada predeterminada para editar la versión neutra. Elige una cultura como el alemán y la superficie de diseño pasa a ser esa versión localizada de la pantalla: sigue mostrando tu diseño, pero cada propiedad que cambies a partir de ahora se registra como una sobrescritura solo para esa cultura.

La disciplina es sencilla: cambia solo lo que debe diferir. En el laboratorio cambias al alemán, traduces las etiquetas, ensanchas los dos botones más largos, mueves una etiqueta y no tocas nada más. Después vuelves al idioma predeterminado antes de cualquier cambio estructural, porque añadir un control con el alemán seleccionado es la forma en que un equipo acaba con una pantalla que tiene un control en un idioma y no en otro.

Más que Text: Size, Location, colores e imágenes

Wisej.NET señala `Size`, `Location`, `Text`, los colores y las imágenes como propiedades que pueden diferir por idioma, y esa lista no es decorativa. Los sustantivos compuestos alemanes son largos, así que un botón dimensionado para *Save* recorta *Speichern*. Los diálogos en francés suelen necesitar más espacio vertical. Una ilustración que se entiende bien en un mercado puede no significar nada en otro, así que una `Image` también es un valor localizado legítimo.

Los caracteres mnemónicos entran en la misma categoría: la etiqueta traducida decide qué letra de acceso rápido está disponible y tiene sentido en ese idioma, así que nunca des por hecho que la inglesa sirve. Y aunque los colores son localizables, cambiarlos por cultura es más raro que cambiar la geometría, así que trata un color por idioma como una decisión que puedas justificar y no como un reflejo.

Qué escribe el diseñador y adónde va

Cuando editas una vista de cultura, el diseñador no añade una rama a `InitializeComponent`. Guarda los valores localizados de las propiedades en archivos de recursos generados junto al contenedor, un conjunto por cada cultura que hayas editado, lo que mantiene los valores específicos de cada idioma fuera del código C# principal del diseñador. Esa separación hace el flujo sostenible: tu código del diseñador sigue siendo legible, los valores por idioma son datos y un revisor puede leer un archivo de recursos sin leer C#.

También traza una frontera para el resto de la aplicación. Los recursos del diseñador pertenecen a una pantalla diseñada, así que las etiquetas fijas, el texto de los botones, los tooltips, las imágenes estáticas y la geometría por idioma encajan ahí. Todo lo que se genera en código, cualquier mensaje que produzca un servicio, cualquier texto de validación y todo lo que reutilice más de una pantalla pertenece a un archivo de recursos de aplicación como `Strings.resx`, leído a través de `Wisej.Resources.ResourceManager`, exactamente lo que construyó el Módulo 1. GlobalDesk mantiene las dos cosas.

Los recursos del diseñador se aplican al crearse los controles

Esta es la regla que sorprende. Los valores localizados en el diseñador se aplican mientras se construye el árbol de controles. Establecer `Application.CurrentCulture` después cambia la cultura de la sesión, y el código que lea recursos a partir de ese momento adopta el nuevo idioma, pero no vuelve atrás a reescribir propiedades en controles que ya existen. Un usuario que cambia de idioma y ve medio editor todavía en inglés se ha topado con esta regla, no con un error.

Tienes dos opciones honestas: recrear el contenedor localizado para que sus valores de recursos se apliquen con la nueva cultura, o conservarlo y escribir código que vuelva a aplicar tú mismo las propiedades respaldadas por recursos. Para una pantalla de este tamaño, recrear es más sencillo, y por eso el laboratorio añade un botón *Recreate editor* que hace *Dispose* del control y construye uno nuevo.

```
private void btnRecreateEditor_Click(object sender, EventArgs e)
{
    // Designer resources are applied while the controls are created,
    // so the new culture only reaches this screen on a rebuild.
    this.hostPanel.Controls.Remove(this.customerEditor);
    this.customerEditor.Dispose();

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

Planificar diseños que sobrevivan a traducciones más largas

Cada sobrescritura de geometría que registras es un valor que alguien tendrá que mantener en todos los idiomas futuros. La sobrescritura más barata es la que nunca hizo falta, así que construye el diseño neutro con elementos que se adapten: acopla un panel en lugar de clavarlo, ancla un botón al borde derecho en lugar de fijar su `Location`, deja que un `FlowLayoutPanel`, un `TableLayoutPanel` o un `FlexLayoutPanel` coloquen una fila, activa `AutoSize` donde una etiqueta deba decidir su propio ancho, y deja padding para que una palabra más larga tenga adónde ir.

Después demuéstalo. Una vista del diseñador no es la respuesta final: ejecuta GlobalDesk en el navegador con la cultura de destino. Busca etiquetas recortadas, botones que se solapan, etiquetas que invaden a su vecina, un orden de tabulación que ya no sigue al orden visual, las imágenes y todos los diálogos que la pantalla puede abrir. El laboratorio termina con una nota que enumera qué propiedades has sobrescrito para el alemán y cuáles has dejado deliberadamente al diseño neutro, y esa segunda lista es la interesante: es la prueba de que el trabajo lo hace tu diseño y no tus sobrescrituras.



Regla de las capas de localización

Decide dónde vive cada cadena antes de teclearla. Los recursos del diseñador pertenecen a una pantalla diseñada y se aplican cuando se crean sus controles; los recursos de aplicación compartidos en `.resx` llevan el texto que genera el código o que reutilizan varias pantallas; `Resources-[LANG].resx` sobrescribe las etiquetas de sistema de Wisej.NET. Nunca guardes una fecha, un número o un importe ya formateados: conserva el valor y dale formato para la cultura activa en el momento de mostrarlo. Y recuerda que la cultura es por sesión, así que el idioma de un usuario nunca debe cambiar la pantalla de otro.



Objetivos de aprendizaje

Al terminar este módulo serás capaz de:

- ✓ Activar la localización en un Form, una Page, un UserControl o un Desktop de Wisej.NET.
- ✓ Usar el selector Language del diseñador para editar propiedades localizadas.
- ✓ Explicar qué propiedades cambian habitualmente por idioma: Text, Size, Location, Color e Images.
- ✓ Entender qué escribe el diseñador en los archivos `.resx`.

- ✓ Planificar diseños que sobrevivan a traducciones más largas.

Guía del laboratorio / examen

Lo que construirás en el laboratorio práctico, el enfoque sugerido y los entregables requeridos.



Laboratorio práctico

Localiza el editor de clientes de GlobalDesk desde el diseñador de Visual Studio. Termina primero el diseño neutro en inglés del control de usuario `CustomerEditor`, después pon `Localizable` en `true` y usa el selector `Language` del diseñador para añadir una variante en alemán, cambiando solo lo que debe diferir: las etiquetas, el ancho de los dos botones más largos y la `Location` de una etiqueta. Ejecuta la aplicación con `de-DE` y compara la salida real del navegador con la vista del diseñador; después añade un botón `Recreate editor` que haga `Dispose` del control y lo reconstruya, para que compruebes por ti mismo por qué los recursos del diseñador no siguen un cambio de cultura en tiempo de ejecución. Termina con una nota de laboratorio que enumere las propiedades que has sobrescrito y las que has dejado deliberadamente al diseño neutro.

Punto de partida

Abres GlobalDesk desde el Módulo 1, donde la página del dashboard ya lee todas sus etiquetas de `Resources/Strings.resx` a través del helper `Texts` y da formato a los valores de su vista previa con `Application.CurrentCulture`. En este laboratorio trabajas sobre el control de usuario `CustomerEditor` que abre el botón `Customers` del dashboard: los campos de código de cliente, nombre, ciudad y notas con sus etiquetas, un group box y los botones `Save` y `Cancel`.

Enfoque sugerido

- ✓ Termina el diseño neutro en inglés de `CustomerEditor` antes de localizar nada: acopla el campo de notas, ancla `Save` y `Cancel` abajo a la derecha, coloca las parejas de etiqueta y editor en un contenedor de diseño como un `TableLayoutPanel` o un `FlexLayoutPanel`, y activa `AutoSize` en las etiquetas que deban ajustarse solas.
- ✓ Selecciona el propio control de usuario en el diseñador, pon `Localizable` en `true` en la ventana `Properties` y confirma que aparece la selección de `Language`; déjala en la entrada predeterminada y recompila una vez para que la versión neutra sea la base.
- ✓ Cambia la selección de `Language` al alemán y traduce solo las etiquetas: las de los campos, la cabecera del group box, `Save` y `Cancel`. Después ensancha mediante su `Size` los dos botones cuyo texto en alemán ya no cabe, y mueve una etiqueta con su `Location` para que la columna siga alineada. Vuelve al idioma predeterminado antes de tocar nada estructural.
- ✓ Ejecuta GlobalDesk con `de-DE` y comprueba el navegador, no el diseñador: busca etiquetas recortadas, controles solapados, una etiqueta que invade a su vecina, el orden de tabulación y los diálogos que puede abrir el editor. Anota todas las diferencias que encuentres entre la vista del diseñador y la página real.
- ✓ Añade un botón **Recreate editor** en la página anfitriona que quite el `CustomerEditor` actual de su contenedor, haga `Dispose` de él y añada uno recién construido. Cambia la cultura, pulsa el botón y observa cómo las etiquetas por fin siguen el cambio; después escribe la nota con cada propiedad que has sobrescrito para el alemán y cada una que has dejado al diseño neutro.

Entregables requeridos

- ✓ Control de usuario CustomerEditor con Localizable en true sobre un diseño neutro en inglés completo
- ✓ Variante de idioma alemán en el diseñador que sobrescribe las etiquetas y al menos un Size y una Location
- ✓ Anchoring, docking o AutoSize aplicados para que las etiquetas en alemán no puedan recortarse
- ✓ Botón Recreate editor que hace Dispose del contenedor y lo reconstruye tras un cambio de cultura
- ✓ Nota de laboratorio con las propiedades sobrescritas y las que se han dejado al diseño neutro

Preguntas de repaso

- ✓ Un compañero añade un campo de teléfono nuevo a `CustomerEditor` con el alemán seleccionado en la lista `Language`. ¿Qué es probable que vea el usuario en inglés y cuál es la regla que lo habría evitado?
- ✓ La confirmación de guardado que muestra el editor tras una actualización correcta sigue viniendo de `Strings.resx` y no de los recursos del diseñador. Da dos razones por las que ese es el sitio adecuado y nombra una etiqueta de esta pantalla que sí pertenezca al diseñador.
- ✓ Tu botón Recreate editor funciona, pero un compañero sostiene que la pantalla debería simplemente volver a aplicar las propiedades localizadas en lugar de reconstruirse. ¿Qué tendría que saber ese código que la reconstrucción no necesita saber, y cuándo sería su enfoque la mejor opción?



Regla de las capas de localización

Decide dónde vive cada cadena antes de teclearla. Los recursos del diseñador pertenecen a una pantalla diseñada y se aplican cuando se crean sus controles; los recursos de aplicación compartidos en `.resx` llevan el texto que genera el código o que reutilizan varias pantallas; `Resources-[LANG].resx` sobrescribe las etiquetas de sistema de Wisej.NET. Nunca guardes una fecha, un número o un importe ya formateados: conserva el valor y dale formato para la cultura activa en el momento de mostrarlo. Y recuerda que la cultura es por sesión, así que el idioma de un usuario nunca debe cambiar la pantalla de otro.