



**Eötvös Loránd Tudományegyetem
Informatikai Kar**

Eseményvezérelt alkalmazások

12. előadás

.NET alkalmazások lokalizációja

Dr. Cserép Máté
mcserep@inf.elte.hu
<https://mcserep.web.elte.hu>

.NET alkalmazások lokalizációja

Lokalizáció

- Általános elvárás, hogy az alkalmazások a felhasználó nyelvén kommunikáljanak, és annak megfelelő nyelvi környezetet használják (pl. számformátum, pénznem, ...), ezt nevezzük az *alkalmazás lokalizációjának* (*application localization*)
- A .NET alkalmazások a lokalizációt egységes formában kezelik (a **System.Globalization** névtérben)
 - az alkalmazás nyelvi környezete (**CultureInfo**) bárhol elérhető, módosítható, illetve az alkalmazás átveheti a rendszer nyelvi környezetét (**CultureInfo.CurrentCulture** és **CultureInfo.CurrentUICulture**)
 - a megjelenített, nyelvfüggő tartalmakat (szöveg, képek) lehetőségünk van lokalizált erőforrásokból (*resource file*) betölteni

.NET alkalmazások lokalizációja

Lokalizáció

- A lokalizációt két lépésben határozhatjuk meg, nyelv, illetve terület (ország szintjén)
 - pl.: angol (Egyesült Államok): **en-US**, angol (Egyesült Királyság): **en-GB**, magyar (Magyarország): **hu-HU**
 - az adott lokalizációhoz kérhetjük le az erőforrásokat (**CultureInfo**), pl.:

```
CultureInfo info = new CultureInfo("en-US");  
// amerikai lokalizáció betöltése
```
 - így lehetőségünk van elérni a lokális információkat, pl.: naptár (**Calendar**), időformátum (**DateTimeFormat**), számformátum (**NumberFormat**), tizedes elválasztó (**NumberFormat.NumberDecimalSeparator**)

.NET alkalmazások lokalizációja

Lokalizált erőforrások

- Az alkalmazás erőforrásait elhelyezhetjük erőforrás fájlokban (.resx), majd közvetlenül, statikus tulajdonságokként hivatkozhatjuk meg őket

- az erőforrás fájl neve az osztálynév, a kulcs a tulajdonság neve lesz, pl.:

```
nameLabel.Text = AppText.NameText;
```

```
// az AppText.resx fájl NameText kulcsú
```

```
// szövegének betöltése, és a címkére állítása
```

- az erőforrásfájlt több nyelvre is elkészíthetjük úgy, hogy a lokalizáció nevét hozzáillesztjük a fájlnevhez, pl.:

```
AppText.en-US.resx
```

```
// szövegek amerikai nyelvi környezetre
```

```
AppText.hu-HU.resx // magyar környezetre
```

.NET alkalmazások lokalizációja

Lokalizált erőforrások

- a rendszer a beállított nyelvi környezetnek megfelelő erőforrásfájl tartalmát tölti be
- amennyiben a lokalizációnak megfelelő fájl nem található, vagy nincs benne megadott kulcsú elem, akkor a lokalizáció nélkül megadott erőforrásfájl tartalmát tölti be
- így célszerű minden esetben egy alapértelmezett erőforrás fájlt is készíteni, pl.:
`AppText.resx // alapértelmezett szövegek`
`AppText.hu-HU.resx`
`// lokalizáció magyar környezetre`
- az erőforrások kezelésére további lehetőségeket (pl. dinamikus betöltés) a **ResourceManager** típus biztosít

.NET alkalmazások lokalizációja

Lokalizált erőforrások a nézetben

- Windows Forms alkalmazások esetén a lokalizációt támogatja a Visual Studio grafikus tervezőfelülete is.
 - A lokalizációt az ablakok (*formok*) szintjén engedélyezhetjük a **Localizable** tulajdonsággal, majd a **Language** tulajdonsággal adhatjuk meg a támogatott nyelveket.
 - A Visual Studio automatikusan elkészíti az ablakhoz tartozó lokalizált erőforrás állományokat.
 - A felületi vezérlők tulajdonságainak kitöltésekor a megfelelő erőforrás állomány frissítésre kerül.
 - A *Designer.cs* állományba olyan C# kód kerül generálásra, amely a *resx* erőforrás állományokból tölti be az értékeket.

.NET alkalmazások lokalizációja

Lokalizált erőforrások XAML nézetben

- WPF és Avalonia alkalmazások esetén amennyiben a lokalizációt nem háttérkódban, hanem a grafikus felületen (XAML-ben) szeretnénk betölteni, a statikus erőforrásokat az **x:Static** hivatkozással tölthetjük be.

- Az ablakon vagy oldalon definiálunk egy névtér kapcsolást a szöveges erőforrásaink helyére:

```
<Window
```

```
...
```

```
xmlns:resx="clr-namespace:MyApp.Resources">
```

- Amit felhasználva betölthetjük a kívánt erőforrást:

```
<Label Text="{x:Static resx:AppText.Test}" />
```

.NET alkalmazások lokalizációja

Példa

Feladat: Készítsünk egy Tic-Tac-Toe programot, amelyben két játékos küzdhet egymás ellen.

- a szövegeket erőforrásban (**ApplicationText**) tároljuk, amelyből készítünk alapértelmezett és magyar változatot
 - ezeket az alkalmazás környezeti rétegben (**App**) és a nézetben (**MainView**) töltjük be
- az alkalmazást Windows Forms, Windows Presentation Foundation és Avalonia UI nézettel is valósítsuk meg, mindhárom változat felületét lokalizáljuk

.NET alkalmazások lokalizációja

Példa

Feladat: Készítsünk egy egyszerű számológép alkalmazást Avalonia UI keretrendszerben, asztali és mobil (Android) környezetre, amellyel a négy alapműveletet végezhethetjük el, illetve láthatjuk korábbi műveleteinket is.

- jelenítsük meg a szövegeket és a tizedes elválasztót a nyelvi beállításnak megfelelően angolul, vagy magyarul
- a tizedes elválasztót a nézetmodell közvetlenül elkérheti a környezettől
- a szövegeket erőforrásban (**ApplicationText**) tároljuk, amelyből készítünk alapértelmezett és magyar változatot
 - ezeket az alkalmazás környezeti rétegben (**App**) és a nézetben (**MainView**) töltjük be