



UWP



User interface



WINDOWS > GUIDES > USER INTERFACE

Localization: Change languages in our UWP viewer

Nutrient Windows SDK comes with many built-in languages:

- ✧ Chinese Simplified / Chinese Traditional (zh-Hans/zh-Hant)
- ✧ Croatian (hr)
- ✧ Czech (cs)
- ✧ Danish (da)
- ✧ Dutch (nl)
- ✧ English (en)
- ✧ English United Kingdom (en-GB)
- ✧ Finnish (fi)
- ✧ French (fr)
- ✧ German (de)
- ✧ Greek (el)
- ✧ Indonesian (id)
- ✧ Italian (it)
- ✧ Japanese (ja)
- ✧ Korean (ko)
- ✧ Malay (ms)



ASK AI

- ✧ Norwegian Bokmål (nb-NO)
- ✧ Polish (pl)
- ✧ Portuguese Brazil / Portugal (pt-BR/pt-PT)
- ✧ Russian (ru)
- ✧ Slovak (sk)
- ✧ Slovenian (sl)
- ✧ Spanish (es)
- ✧ Swedish (sv)
- ✧ Thai (th)
- ✧ Turkish (tr)
- ✧ Ukrainian (uk)
- ✧ Welsh (cy)

Forcing a specific language

By default, the `PdfView` localization will mirror the `CurrentUICulture` of your UWP application. There are two ways of overriding this default locale.

The first way is through the `ViewState` when showing a new document:

```
1  await PdfView.Controller.ShowDocumentWithViewStateAsync(documentSource, new  
2  {  
3      Locale = "nl"  
4  });
```

The second way is at runtime, through the `Controller`, after the UI is displayed:

```
1  PdfView.OnDocumentOpened += async (pdfView, document) =>  
2  {  
3      await pdfView.Controller.SetLocaleAsync("pt-BR");  
4  };
```

If either your application language or the language specified are not supported, the PdfView will automatically detect the closest language. If it is unable to, the locale will default to English.

To get a list of supported locales at runtime, you can use the following:

```
await PDFView.Controller.GetSupportedLocalesAsync();
```

The current PdfView locale code is also available:

```
await PDFView.Controller.GetCurrentLocaleAsync();
```

Was this helpful?

☒ YES

☐ NO

Questions? [Contact us](#)

