

WPF for .NET 9 で 変わったところを 見てみよう！

とっちゃん(高萩 俊行)

Microsoft MVP for Developer Technologies

Since 2005/10-

<https://github.com/Tocchann/FluentDemo>



.NETラボ 勉強会 2025年 3月

アジェンダ

- 自己紹介
- WPF for .NET 9 の主な変更点
- C#13で追加された field キーワード(Preview)
- まとめ

とっちゃん(高萩 俊行)

Microsoft MVP for Developer Technologies

Since 2005/10-

<https://github.com/Tocchann/FluentDemo>



.NETラボ 勉強会 2025年 3月

自己紹介

- とっちゃん
 - 高萩 俊行
 - わんくま同盟
 - Microsoft MVP
 - 2005/10–2008/09
 - Windows SDK
 - 2008/10-2016/09
 - Visual C++
 - 2016/10-2018
 - Visual Studio and Development Technologies
 - 2018-
 - Developer Technologies
- 市販パッケージソフト開発
 - 昔ながらのISV
 - Independent Software Vender
 - 良くも悪くも昭和のソフトハウス
 - 絶賛リモートワーク中

• 会社員

<https://github.com/Tocchann/FluentDemo>



アジェンダ

- WPF for .NET 9 の主な変更点
- C#13で追加された field キーワード(Preview)
- まとめ

UIの変更はインパクトが大きい！
標準搭載機能の追従がベストとは言えない！

<https://github.com/Tocchann/FluentDemo>



WPF for .NET 9 の主な変更点

- Fluent UI リソースを定義
- Windows アクセントカラーのサポート
- ハイフンベースの合字サポート
- ThemeMode の導入
- Fluent UI テーマの組み込み対応
- BinaryFormatter のサポート終了

詳しくは WPF for .NET 9 の新機能を参照

<https://github.com/Tocchann/FluentDemo>



Fluent UI リソースを定義

- ThemeModeを指定するか、リソースに追加する
 - リソースに追加しないと一部うまく反映されない場合がある
- ResourceDictionaryはApp.xaml に追加
 - 追加した場合、FluentUIはOFFにならない
 - VS2022 (17.13)のデザイナーモードはまだ対応不十分
 - デザイン時に ThemeMode をうまく反映できない

```
<ResourceDictionary>  
  <ResourceDictionary.MergedDictionaries>  
    <ResourceDictionary  
      Source="pack://application:,,,/PresentationFramework.Fluent;component/Themes/Fluent.xaml" />  
    </ResourceDictionary.MergedDictionaries>  
  </ResourceDictionary>
```



Windows アクセントカラーのサポート

- スタイル、カラーとして各種のアクセントカラーを用意
- DynamicResource なので動的変動に追従
- コントロールのスタイルとして用意

ハイフンベースの合字サポート

- FontFamily に Cascadia Code を指定
 - Fluent UI とは無関係に利用可能
- 日本語と組み合わせる時は注意が必要

```
<TextBlock Text="&lt;- &lt;-- &lt;== ==&gt; --&gt;" FontFamily="Cascadia Code" />  
<TextBlock Text="-&gt; != &lt;= &gt;= /= *= += -= " FontFamily="Cascadia Code" />
```

← ←← ⇐ ⇒ →
→ ≠ ≤ ≥ ≯ *= += -=

ThemeMode の導入

- ThemeMode.System
 - システム状態に追従
- ThemeMode.Light
 - 常にライトモード
- ThemeMode.Dark
 - 常にダークモード
- ThemeMode.None
 - 旧UI(Aero2)
 - ResourceDictionary を追加している場合はSystemと同じ

Fluent UI テーマの組み込み対応

- 見た目の変化に伴う変更点
- 使えなくなったモード
- ボタンレイアウトの変更

見た目の変化に伴う変更点

- タップされる可能性のあるコンテンツのPaddingの増量
 - Padding(とくに高さ部分)が従来のLabelと同等程度に
- 無効状態がテキストも対象になったものがある
- RadioButton/CheckBox が大きくなった
 - 標準テキストサイズならボタン部分のほうが大きい

使えなくなったモード

- GridView(ListView.View)が使えない
 - 今後(.NET10)サポートされるようになるかは不明
- 標準コントロールでの代替手段は DataGrid くらい
 - DataGrid.IsReadOnly=Trueなら編集はできない
 - 見た目が変わる(ListViewではないし、選択状態も違う)
 - ResourceDictionary の有無で見た目が変わってしまう
- GridView の代替として動的切り替えはできない
 - コントロールを作り直せばOKだと思うが未確認

ボタンレイアウトの変更

- ThemeMode == None(従来)
 - 提供領域を全て利用する
- ThemeMode != None(Fluent UI)
 - ボタンコンテンツの大きさが最大値
 - 縦方向は中央寄せ、横方向は左寄せ
 - Gridパネルで領域計算していると想定外の位置に来る
 - HorizontalAlignment 効果あり
 - HorizontalContentAlignment 効果なし(常に左寄せ)
 - VerticalAlignment 効果なし(常にセンター)
 - VerticalContentAlignment 効果なし(常にセンター)



DEMO

見た目の変化を実感しよう！



.NETラボ 勉強会 2025年 3月

BinaryFormatter のサポート終了

- 逆シリアル(Deserialize)攻撃の脆弱性回避
 - サービス拒否
 - 情報漏洩
 - リモートコードの実行
 - e.t.c.
- DataObject の読み書きで影響
 - DataFormats.Serializableや独自形式はNG
 - byte[] や Stringペイロード(JSONなど)に変更
- 詳細は[WPF の BinaryFormatter 移行ガイド](#)



C#13で追加された field キーワード(Preview)

- プロパティアクセッサのバックイングフィールドとして追加
- LangVersion を preview にする必要がある
- 主に自動生成のために作られている
- CommunityToolkit.Mvvm/M(8.4.0以上)で利用

```
public int Property
{
    get => field;
    set
    {
        if (field != value){
            field = value;
            PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(nameof(Property)));
        }
    }
}
```



まとめ

- 標準でUIモードに選択肢ができたのは初めて
- FluentUIモードのON/OFF切り替えは実質出来ない
 - Windowsの機能による描画ではない
 - ResourceDictionary を追加せずに使えるなら可能
- 実質的にプレビュー状態
 - .NET 10 でブレイクチェンジがある可能性は大きい
 - UIライブラリの対応はこれからと思われる
 - 次に情報がでるとしたら、5月開催のBuild
 - WinUIとの連動は今後の課題

まとめ

- 参考情報

- [WPF for .NET 9](#)
- [WPF Gallery Preview](#)
- [WPFのBinaryFormatter移行ガイド](#)
- [.NET 9 の新機能](#)
- [C# 13 の新機能](#)

UIの変更はインパクトが大きい！
標準搭載機能の追従がベストとは言えない！