

مطلب « [نحوه نمایش تمام آیکون‌های تعریف شده در یک قلم در WPF](#) » را در نظر بگیرید. سؤال: اگر در یک برنامه تنها به تعدادی از این آیکون‌ها یا گلیف‌ها نیاز بود آیا می‌توان این‌ها را به صورت مجزا استخراج و استفاده کرد؟

پاسخ: بلی. همان کلاس FontFamily موجود در اسمبلی PresentationCore.dll، امکان تبدیل یک گلیف را به معادل هندسی آن نیز دارد. در ادامه کدهای آن را مرور خواهیم کرد:

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.IO;
using System.Windows;
using System.Windows.Media;
using CrMap.Models;

namespace CrMap.ViewModels
{
    public class CrMapViewModel
    {
        public IList<Symbol> Symbols { set; get; }
        public int GridRows { set; get; }
        public int GridCols { set; get; }

        public CrMapViewModel()
        {
            fillDataSource();
        }

        private void fillDataSource()
        {
            Symbols = new List<Symbol>();
            GridCols = 15;

            var fontFamily = new FontFamily(new Uri("pack://application:,,,/"), "/Fonts/#whhglyphs");

            GlyphTypeface glyph = null;
            Typeface glyphTypeface = null;
            foreach (var typeface in fontFamily.GetTypefaces())
            {
                if (typeface.TryGetGlyphTypeface(out glyph) && (glyph != null))
                {
                    glyphTypeface = typeface;
                    break;
                }
            }

            if (glyph == null)
                throw new InvalidOperationException("Couldn't find a GlyphTypeface.");

            GridRows = (glyph.CharacterToGlyphMap.Count / GridCols) + 1;

            foreach (var item in glyph.CharacterToGlyphMap)
            {
                var index = item.Key;
                Symbols.Add(new Symbol
                {
                    Character = Convert.ToChar(index),
                    CharacterCode = string.Format("&#x{0:X}", index)
                });

                saveToFile(glyphTypeface, index);
            }
        }

        private static void saveToFile(Typeface glyphTypeface, int index)
        {
            var formattedText = new FormattedText(
                textToFormat: Convert.ToChar(index).ToString(),
                culture: new CultureInfo("en-us"),
                flowDirection: FlowDirection.LeftToRight,
                typeface: glyphTypeface,
                emSize: 20,
```

```

        foreground: Brushes.Black);
var geometry = formattedText.BuildGeometry(new Point(0, 0));
var path = geometry.GetFlattenedPathGeometry();
File.WriteAllText(index + ".path", path.ToString());
    }
}
}

```

در اینجا تنها متد saveToFile در مقایسه با قسمت قبل افزوده شده است.

شیء FormattedText دارای متدی است به نام BuildGeometry که اطلاعات یک گلیف را تبدیل به معادل هندسی آن می‌کند. سپس توسط متد GetFlattenedPathGeometry معادل Path آن را می‌توان بدست آورد. برای مثال اگر پس از اجرای این مثال، به فایل path.48 تولیدی آن مراجعه کنیم، چنین خروجی را می‌توان مشاهده کرد:

```

F1M5,7.47150993347168L5,17.2566661834717 5.732421875,19.0242443084717 7.5,19.7566661834717
12.5,19.7566661834717 13.69140625,19.4441661834717 5,7.47150993347168z
M7.5,4.75666618347168L6.30859375,5.06916618347168 15,17.0418224334717
15,7.25666618347168 14.267578125,5.48908805847168 12.5,4.75666618347168
7.5,4.75666618347168z M7.5,2.25666618347168L12.5,2.25666618347168
14.4189453125,2.62287712097168 16.03515625,3.72150993347168 17.1337890625,5.33772134780884
17.5,7.25666618347168 17.5,17.2566661834717 17.1337890625,19.1756114959717
16.03515625,20.7918224334717 14.4189453125,21.8904552459717 12.5,22.2566661834717
7.5,22.2566661834717 5.5810546875,21.8904552459717 3.96484375,20.7918224334717
2.8662109375,19.1756114959717 2.5,17.2566661834717 2.5,7.25666618347168 2.8662109375,5.33772134780884
3.96484375,3.72150993347168 5.5810546875,2.62287712097168 7.5,2.25666618347168z

```

که برای استفاده از اطلاعات آن در WPF می‌توان نوشت:

```

<Path Stroke="DarkRed" Fill="Black" Data="F1M5,7.47150993347168L5,17.2566661834717
5.732421875,19.0242443084717 7.5,19.7566661834717 12.5,19.7566661834717
13.69140625,19.4441661834717
5,7.47150993347168z M7.5,4.75666618347168L6.30859375,5.06916618347168 15,17.0418224334717
15,7.25666618347168 14.267578125,5.48908805847168 12.5,4.75666618347168
7.5,4.75666618347168z
M7.5,2.25666618347168L12.5,2.25666618347168 14.4189453125,2.62287712097168
16.03515625,3.72150993347168 17.1337890625,5.33772134780884 17.5,7.25666618347168
17.5,17.2566661834717 17.1337890625,19.1756114959717 16.03515625,20.7918224334717
14.4189453125,21.8904552459717 12.5,22.2566661834717 7.5,22.2566661834717
5.5810546875,21.8904552459717 3.96484375,20.7918224334717 2.8662109375,19.1756114959717
2.5,17.2566661834717 2.5,7.25666618347168 2.8662109375,5.33772134780884
3.96484375,3.72150993347168
5.5810546875,2.62287712097168 7.5,2.25666618347168z" />

```