

در ویندوز XP زمانیکه زبان سیستم و همچنین کشور جاری به ایران تنظیم شود، VS.Net فایل‌های ANSI را از نوع ANSI-Windows 1256 (یا همان ANSI-Arabic) در نظر می‌گیرد و مشکلی هم برای ذخیره داده‌های یونیکد در این نوع فایل‌های ANSI ویژه نخواهد بود (الزامی وجود ندارد که این فایل‌ها حتما به فرمت UTF8 ذخیره شوند). اما در ویندوز 7 با همان تنظیمات، VS.Net این فایل‌ها را با encoding از نوع windows-1252 تشخیص می‌دهد و پس از کامپایل برنامه‌ای که قبلا مشکل نداشت، این بار همه چیز به همه ریخته خواهد بود. شاید اینطور به نظر برسد که این فایل‌ها خراب شده‌اند، ولی خیر. مشکلی وجود ندارد؛ فقط فرمت encoding خواندن آن‌ها باید windows-1256 باشد (و نه 1252) و گرنه تخریب شده به نظر می‌رسند.

```
protected void Page_Load(object sender, EventArgs e)
{
    Page.Title = "ÊÚŃiÝ ãİiÑçà";
}
```

تعداد فایل‌ها هم زیاد است و نیاز به یک روش سریع برای رفع این مشکل وجود داشت. بنابراین سه عملیات باید صورت گیرد:

لیست کردن تمام فایل‌های مورد نظر (فایل‌های cs و aspx و امثال آن فقط)

پیدا کردن encoding جاری فایل‌ها : کدامیک از آن‌ها با فرمت UTF-8 ذخیره نشده‌اند؟

تبدیل به یونیکد: خواندن این فایل‌های غیر یونیکد یافت شده با فرمت windows-1256 و سپس ذخیره مجدد با فرمت UTF-8

که خلاصه روش انجام کار به صورت زیر است:

الف) آیا فایل جاری مورد نظر با فرمت UTF-8 with signature ذخیره شده است؟

این signature در مورد فایل‌های UTF-8 به سه بایت اول فایل بر می‌گردد که اصطلاحا [byte-order mark](#) یا BOM گفته می‌شود و باید مساوی EFBBBF باشد. چون فایل‌های ANSI این امضا را ندارند، در یک سیستم ممکن است 1256 خوانده شوند و در یک سیستم دیگر 1252 یا نوع‌های ANSI دیگر بسته به تنظیمات جاری سیستم و مشکل اصلی از VS.Net نیست.

```
/// <summary>
/// آیا فایل مورد نظر با فرمت یونیکد دارای امضا ذخیره شده است؟
/// </summary>
/// <param name="filePath">ورودی فایل</param>
/// <returns>بله یا خیر</returns>
public static bool IsUTF8(string filePath)
{
    using (FileStream file = new FileStream(filePath,
        FileMode.Open, FileAccess.Read, FileShare.Read))
    {
        if (file.CanSeek)
        {
            byte[] bom = new byte[4]; // Get the byte-order mark, if there is one
            file.Read(bom, 0, 4);
            if ((bom[0] == 0xef && bom[1] == 0xbb && bom[2] == 0xbf)) // utf-8
            {
                return true;
            }
            else
            {
                return false;
            }
        }
    }
}
```

```

    }
    else
    {
        //احتمالا فایل بایناری است
        return false;
    }
}
}

```

ب) خواندن یک فایل ANSI عربی با فرمت windows-1256 بدون تخریب اطلاعات آن و سپس ذخیره سازی با فرمت UTF-8

```

/// <summary>
/// تبدیل یک فایل انسی عربی به یونیکد دارای امضاء
/// </summary>
/// <param name="path">مسیر ورودی</param>
public static void FixWindows1256(string path)
{
    try
    {
        //باز کردن فایل با فرمت انسی عربی و تبدیل به یونیکد
        string data = File.ReadAllText(path, Encoding.GetEncoding("windows-1256"));
        //نوشتن حاصل یونیکد در جای قبلی با فرمت مربوطه
        File.WriteAllText(path, data, Encoding.UTF8);
    }
    catch (Exception ex)
    {
        //دسترسی وجود ندارد یا امثال آن
        Console.WriteLine(ex.ToString());
    }
}

```

پ.ن.

جالب اینجا است که این نوع فایل‌های ANSI عربی در وب زیاد پیدا می‌شوند. برای مثال [اینجا](#) کلیک کنید. تمام این نوع فایل‌ها را با متد فوق می‌توان بدون تخریب اطلاعات به فرمت UTF-8 دارای امضاء اصلاح کرد.

نظرات خوانندگان

نویسنده: Hossein Moradinia
تاریخ: ۱۷:۳۱:۱۸ ۱۳۸۸/۰۹/۲۰

بازم تشکر به خاطر زحمت زیادی که می کشید

نویسنده: MDP
تاریخ: ۱۹:۱۶:۴۸ ۱۳۸۸/۰۹/۲۰

سلام خسته نباشید.

منم 3 روز سوییچ کردم روی ویندوز سون.با فارسی یکم مشکل داره.

خوشحال میشم راه حل های مشکلاتی که بر می خورید رو بلاگ کنید.صد در صد استفاده های زیادی خواهد داشت.