

عنوان: تغییر نام دسته جمعی تعدادی فایل PDF بر اساس متادیتای فایل‌ها

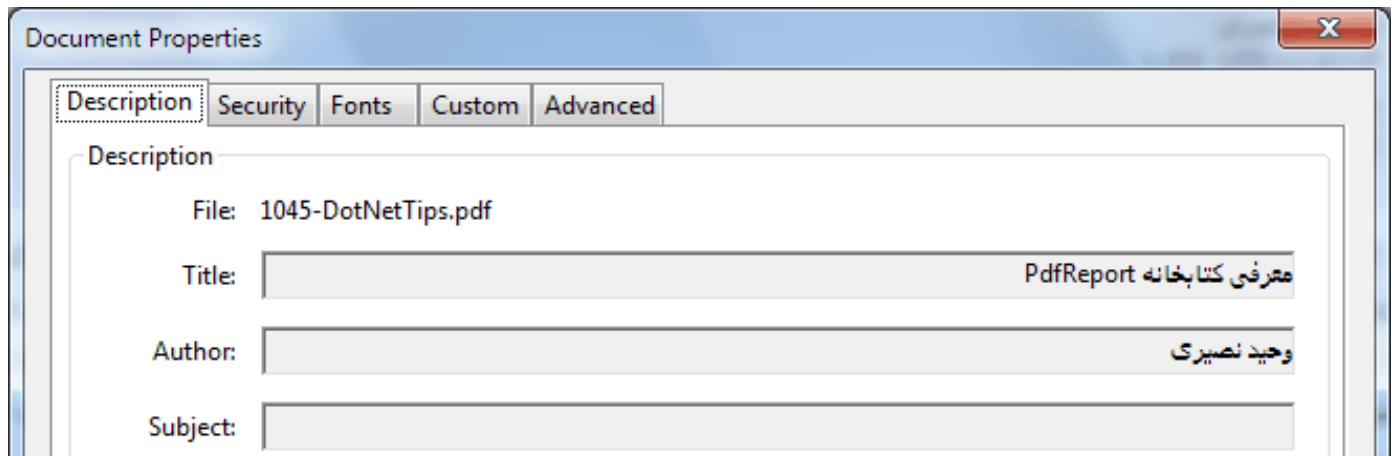
نویسنده: وحید نصیری

تاریخ: ۸:۵۱۳۹۱/۱۰/۲۳

آدرس: www.dotnettips.info

برچسب‌ها: iTextSharp, PDF

فرض کنید تعداد زیادی فایل PDF را با اسامی نامفهومی داریم. برای نظم بخشیدن و یافتن ساده‌تر مطالب شاید بهتر باشد این فایل‌ها را بر اساس عنوان اصلی ذخیره شده در فایل، تغییر نام دهیم.



امکان خواندن meta data فوق (البته در صورت وجود)، توسط iTextSharp وجود دارد. در ادامه قطعه کد ساده‌ای را ملاحظه می‌کنید که در یک پوشه، تمام فایل‌های PDF را یافته و بر اساس Title یا Subject آن‌ها، فایل موجود را تغییر نام می‌دهد:

```
using System.IO;
using iTextSharp.text.pdf;

namespace BatchRename
{
    class Program
    {
        private static string getTitle(PdfReader reader)
        {
            string title;
            reader.Info.TryGetValue("Title", out title); // Reading PDF file's meta data
            return string.IsNullOrEmpty(title) ? string.Empty : title.Trim();
        }

        private static string getSubject(PdfReader reader)
        {
            string subject;
            reader.Info.TryGetValue("Subject", out subject); // Reading PDF file's meta data
            return string.IsNullOrEmpty(subject) ? string.Empty : subject.Trim();
        }

        static void Main(string[] args)
        {
            var dir = @"D:\Path";
            if (!dir.EndsWith(@"\"))
                dir = dir + @"\";

            foreach (var file in Directory.GetFiles(dir, "*.pdf"))
            {
                var reader = new PdfReader(file);
                var title = getTitle(reader);
                var subject = getSubject(reader);
                reader.Close();

                string newFile = string.Empty;
                if (!string.IsNullOrEmpty(title))
                {
                    newFile = dir + title + ".pdf";
                }
            }
        }
    }
}
```

```

    }
    else if (!string.IsNullOrEmpty(subject))
    {
        newFile = dir + subject + ".pdf";
    }

    if (!string.IsNullOrEmpty(newFile))
        File.Move(file, newFile);
    }
}
}
}

```



در قطعه کد فوق علت مراجعه به reader.Info، بر اساس ساختار یک فایل PDF است. در Dictionary به نام Info (تصویر فوق)، در یک سری کلید مشخص، اطلاعاتی مانند تهیه کننده، عنوان و غیره درج می‌شوند. به این ترتیب با استفاده از شیء PdfReader، فایل را گشوده، این متادیتا را خوانده و سپس بر اساس آن می‌توان فایل را تغییر نام داد.

نظرات خوانندگان

نویسنده: علیرضا نوری
تاریخ: ۱۳۹۱/۱۰/۲۴ ۲:۳۱

با وجود اینکه می‌دونم برای سادگی خیلی مسائل رو در نظر نگرفتید، اما چون این کد برای آموزش، بهتره که خوب نوشته بشه. برای مثال Concat کردن چند رشته در Net. کار درستی نیست. علاوه بر اون، بهتره که از تابع Path.Combine برای اتصال دو مسیر استفاده بشه.
مرسی

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۴ ۱۰:۵۵

[بستگی داره](#) چه تعداد رشته رو قراره با هم جمع بزنید. اگر 10 هزار مورد است، استفاده از StringBuilder می‌تونه مفید باشه؛ اما اگر فقط سه قطعه است (مانند تشکیل newFile در مثال بالا)، تفاوتی را در کارایی [احساس نخواهید کرد](#)؛ ضمن اینکه سرشار تولید و وهله سازی StringBuilder برای اتصال فقط سه قطعه با هم می‌تونه تا چهار برابر حالت‌های معمولی باشه (مراجعه کنید به قسمت ابتدای نمودار مقاله [Performance considerations for strings](#)؛ در این نمودار، تفاوت پس از اتصال 600 قطعه به هم، خودش رو نشون می‌ده).

