

یکی از شروط تهیه آزمون‌های واحد، خارج نشدن از مرزهای سیستم در حین بررسی آزمون‌های مورد نظر است؛ تا بتوان تمام آزمون‌ها را با سرعت بسیار بالایی، بدون نگرانی از در دسترس نبودن منابع خارجی، درست در لحظه انجام آزمون‌ها، به پایان رساند. اگر این خروج صورت گیرد، بجای unit tests با integration tests سر و کار خواهیم داشت. در این میان، کار با فایل‌ها نیز مصداق بارز خروج از مرزهای سیستم است.

برای حل این مشکل راه حل‌های زیادی توصیه شده‌اند؛ منجمله تهیه یک اینترفیس محصور کننده فضای نام System.IO و سپس استفاده از فریم ورک‌های mocking و امثال آن. یک نمونه از پیاده سازی آن را اینجا می‌توانید پیدا کنید: (+) اما راه حل ساده‌تری نیز برای این مساله وجود دارد و آن هم افزودن فایل‌های مورد نظر به پروژه آزمون واحد جاری و سپس مراجعه به خواص فایل‌ها و تغییر Build Action آن‌ها به Embedded Resource می‌باشد. به این صورت پس از کامپایل پروژه، فایل‌های ما در قسمت منابع اسمبلی جاری قرار گرفته و به کمک متد زیر قابل دسترسی خواهند بود:

```
using System.IO;
using System.Reflection;

public class UtHelper
{
    public static string GetInputFile(string filename)
    {
        var thisAssembly = Assembly.GetExecutingAssembly();
        var stream = thisAssembly.GetManifestResourceStream(filename);
        return new StreamReader(stream).ReadToEnd();
    }
}
```

نکته‌ای را که اینجا باید به آن دقت داشت، filename متد GetInputFile است. چون این فایل دیگر به صورت متداول از فایل سیستم خوانده نخواهد شد، نام واقعی آن به صورت namespace.filename می‌باشد (همان نام منبع اسمبلی جاری). اگر جهت یافتن این نام با مشکل مواجه شدید، تنها کافی است اسمبلی آزمون واحد را با برنامه Reflector یا ابزارهای مشابه گشوده و نام منابع آن را بررسی کنید.