عنوان: **تولید MiniDump در حین کرش برنامه‌های دات نت**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۱۶ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: Debugging, exception handling, Crash Dump

با مطالعه‌ی سورس‌های محصولات اخیرا سورس باز شده‌ی مایکروسافت، نکات جالبی را می‌توان استخراج کرد. برای نمونه اگر سورس پروژه‌ی Orleans را بررسی کنیم، در حین بررسی اطلاعات استثناءهای رخ داده‌ی در برنامه، متد TraceLogger.CreateMiniDump نیز بکار رفته‌است. در این مطلب قصد داریم، این متد و نحوه‌ی استفاده‌ی از حاصل آن‌را بررسی کنیم.

تولید MiniDump در برنامه‌های دات نت

خلاصه‌ی روش تولید MiniDump در پروژه‌ی Orleans به صورت زیر است:

الف) حالت‌های مختلف تولید فایل دامپ که مقادیر آن قابلیت Or شدن را دارا هستند:

```
[Flags]
public enum MiniDumpType
{
    MiniDumpNormal = 0x00000000,
    MiniDumpWithDataSegs = 0x00000001,
    MiniDumpWithFullMemory = 0x00000002,
    MiniDumpWithHandleData = 0x00000004,
    MiniDumpFilterMemory = 0x00000008,
    MiniDumpScanMemory = 0x00000010,
    MiniDumpWithUnloadedModules = 0x00000020,
    MiniDumpWithIndirectlyReferencedMemory = 0x00000040,
    MiniDumpFilterModulePaths = 0x00000080,
    MiniDumpWithProcessThreadData = 0x00000100,
    MiniDumpWithPrivateReadWriteMemory = 0x00000200,
    MiniDumpWithoutOptionalData = 0x00000400,
    MiniDumpWithFullMemoryInfo = 0x00000800,
    MiniDumpWithThreadInfo = 0x00001000,
    MiniDumpWithCodeSegs = 0x00002000,
    MiniDumpWithoutManagedState = 0x00004000
}
```

ب) متد توکار ویندوز برای تولید فایل دامپ

```
public static class NativeMethods
{
    [DllImport("Dbghelp.dll")]
    public static extern bool MiniDumpWriteDump(
        IntPtr hProcess,
        int processId,
        IntPtr hFile,
        MiniDumpType dumpType,
        IntPtr exceptionParam,
        IntPtr userStreamParam,
        IntPtr callbackParam);
}
```

ج) فراخوانی متد تولید دامپ در برنامه

در اینجا نحوه‌ی استفاده از enum و متد MiniDumpWriteDump ویندوز را مشاهده می‌کنید:

```
public static class DebugInfo
{
    public static void CreateMiniDump(
        string dumpFileName, MiniDumpType dumpType = MiniDumpType.MiniDumpNormal)
    {
        using (var stream = File.Create(dumpFileName))
        {
            var process = Process.GetCurrentProcess();
            // It is safe to call DangerousGetHandle() here because the process is already crashing.
            NativeMethods.MiniDumpWriteDump(
                        process.Handle,
```

```
                            process.Id,
                            stream.SafeFileHandle.DangerousGetHandle(),
                            dumpType,
                            IntPtr.Zero,
                            IntPtr.Zero,
                            IntPtr.Zero);
        }
    }

    public static void CreateMiniDump(MiniDumpType dumpType = MiniDumpType.MiniDumpNormal)
    {
        const string dateFormat = "yyyy-MM-dd-HH-mm-ss-fffZ"; // Example: 2010-09-02-09-50-43-341Z
        var thisAssembly = Assembly.GetEntryAssembly() ?? Assembly.GetCallingAssembly();
        var dumpFileName = string.Format(@"{0}-MiniDump-{1}.dmp",
                    thisAssembly.GetName().Name,
                    DateTime.UtcNow.ToString(dateFormat, CultureInfo.InvariantCulture));

        var path = Path.Combine(getApplicationPath(), dumpFileName);
        CreateMiniDump(path, dumpType);
    }

    private static string getApplicationPath()
    {
        return HttpContext.Current != null ?
            HttpRuntime.AppDomainAppPath :
            Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location);
    }
}
```

متد MiniDumpWriteDump نیاز به اطلاعات پروسه‌ی جاری، به همراه هندل فایلی که قرار است اطلاعات را در آن بنویسد، دارد. همچنین dump type آن نیز می‌تواند ترکیبی از مقادیر enum مرتبط باشد.

یک مثال:

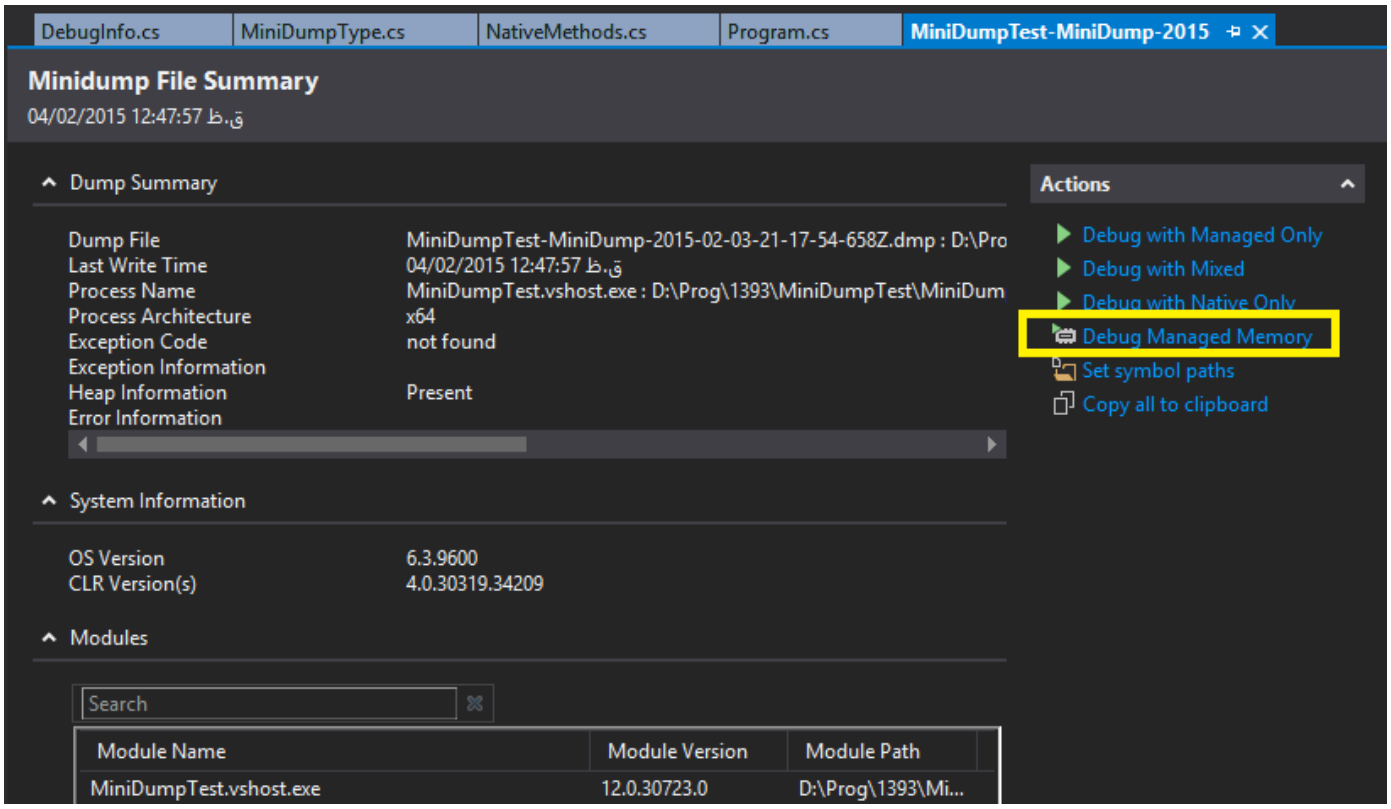
```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            var zero = 0;
            Console.WriteLine(1 / zero);
        }
        catch (Exception ex)
        {
            Console.Write(ex);
            DebugInfo.CreateMiniDump(dumpType:
                                MiniDumpType.MiniDumpNormal |
                                MiniDumpType.MiniDumpWithPrivateReadWriteMemory |
                                MiniDumpType.MiniDumpWithDataSegs |
                                MiniDumpType.MiniDumpWithHandleData |
                                MiniDumpType.MiniDumpWithFullMemoryInfo |
                                MiniDumpType.MiniDumpWithThreadInfo |
                                MiniDumpType.MiniDumpWithUnloadedModules);

            throw;
        }
    }
}
```

در اینجا نحوه‌ی فراخوانی متد CreateMiniDump را در حین کرش برنامه مشاهده می‌کنید. پارامترهای اضافی دیگر سبب خواهند شد تا اطلاعات بیشتری از حافظه‌ی جاری سیستم، در دامپ نهایی قرار گیرند. اگر پس از اجرای برنامه، به پوشه‌ی bin\debug مراجعه کنید، فایل dmp تولیدی را مشاهده خواهید کرد.

نحوه‌ی بررسی فایل‌های dump

الف) با استفاده از Visual studio 2013

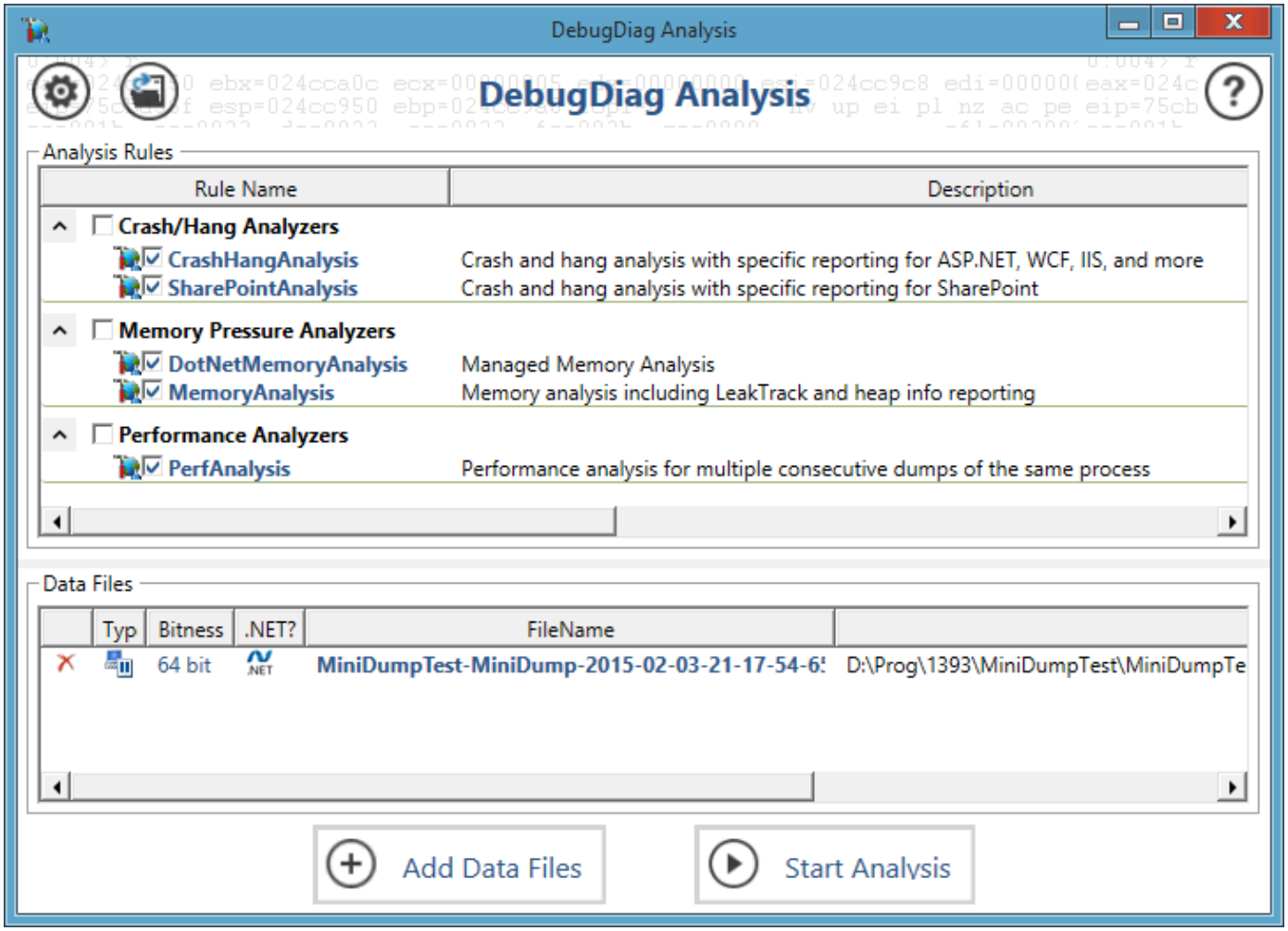
از به روز رسانی سوم VS 2013 به بعد، فایل‌های dump را می‌توان داخل خود VS.NET نیز آنالیز کرد (^ و ^ و ^). برای نمونه تصویر ذیل، حاصل کشودن فایل کرش مثال فوق است:



در اینجا اگر بر روی لینک debug managed memory کلیک کنید، پس از چند لحظه، آنالیز کامل اشیاء موجود در حافظه را در حین تهیه‌ی دامپ تولیدی، می‌توان مشاهده کرد. این مورد برای آنالیز نشتی‌های حافظه‌ی یک برنامه بسیار مفید است.

ب) استفاده از برنامه‌ی Debug Diagnostic Tool

برنامه‌ی Debug Diagnostic Tool را از اینجا می‌توانید دریافت کنید. این برنامه نیز قابلیت آنالیز فایل‌های دامپ را داشته و اطلاعات بیشتری را پس از آنالیز ارائه می‌دهد.



برای نمونه پس از آنالیز فایل دامپ تهیه شده توسط این برنامه، خروجی ذیل حاصل می‌شود:

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید:

MiniDumpTest.zip