**Windbg内存分析案例**

目录

[1. 简介 1](#_Toc17253_WPSOffice_Level1)

[2. 搭建环境 1](#_Toc8745_WPSOffice_Level1)

[3. 内存案例分析 1](#_Toc267_WPSOffice_Level1)

[4. 学习资源 3](#_Toc5893_WPSOffice_Level1)

1. **简介**

1)SOS 调试扩展 (SOS.dll) 通过提供有关内部公共语言运行时 (CLR) 环境的信息，帮助你在 Visual Studio 和 Windows 调试器 (WinDbg.exe) 中调试托管程序。 此工具需要你启用项目的非托管调试。 SOS.dll 自动随 .NET Framework 一起安装。

2)Visual Studio中没有学，大家有兴趣可以学习并分享一下

3)简单的分享,带大家入个门

4)下面的环境搭建我使用了一种我常用的方式，其他方法大家可以去独自学习。

1. **搭建环境**

1)安装windbg软件

2)获取dump文件,方式：跟运维获取

3)获取对应项目的pdb文件，放到文件夹中

4)使用windbg打开dump文件

5)方式：File->Open Crush Dump

6)使用命令加载pdb文件

1>制定路径 .sympath D:\学习代码\SymbolFile\MicrosoftSymbol\*SRV\*http://msdl.microsoft.com/download/symbols;D:\北森\学习代码\SymbolFile\360.tms.beisen.com

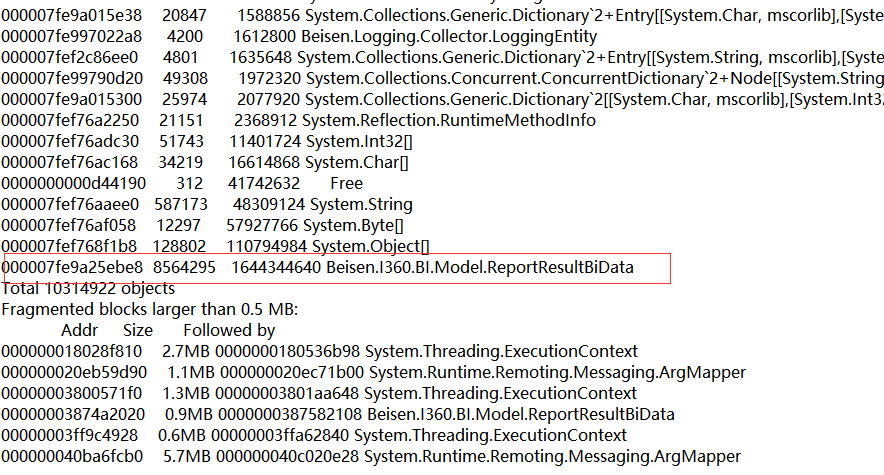
2>加载pdb .symfix

3>.reload

4>.loadby sos clr

1. **内存案例分析**

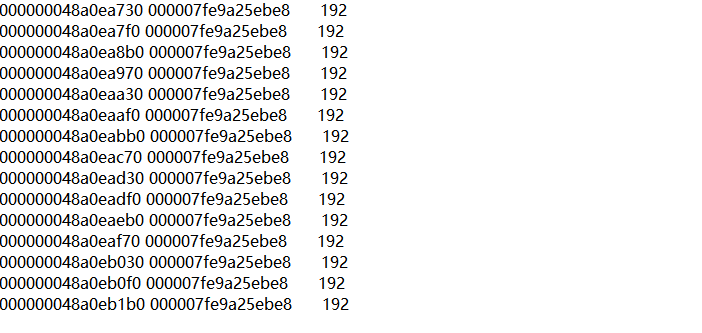
1.使用!Dumpheap命令查看堆内存

如图：

2.!dumpheap -mt [MT]

1红框中的000007fe9a25ebe8为MT

功能是：查看MT中对象地址

执行结果：

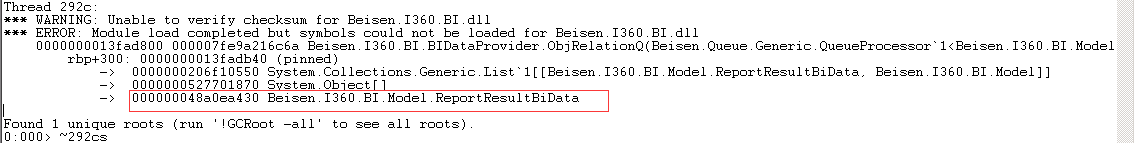
第一列为对象地址(address)

3.!do [address]

功能：这个内存查看具体的对象

4.!gcroot [address]

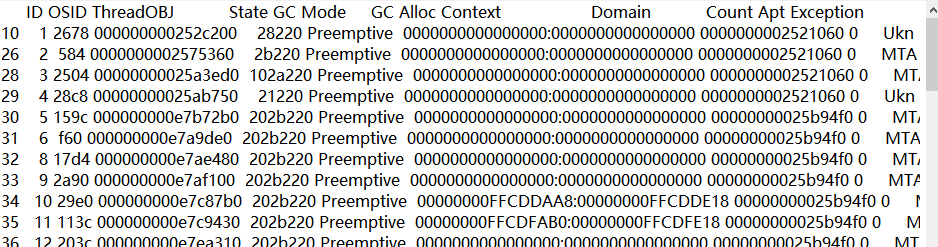
功能：查看对象引用

效果图：

上图能看出Thread 292c，292c为下面命令中的ThreadOBJ列

5.!threads

功能：查看工作的线程

效果图：

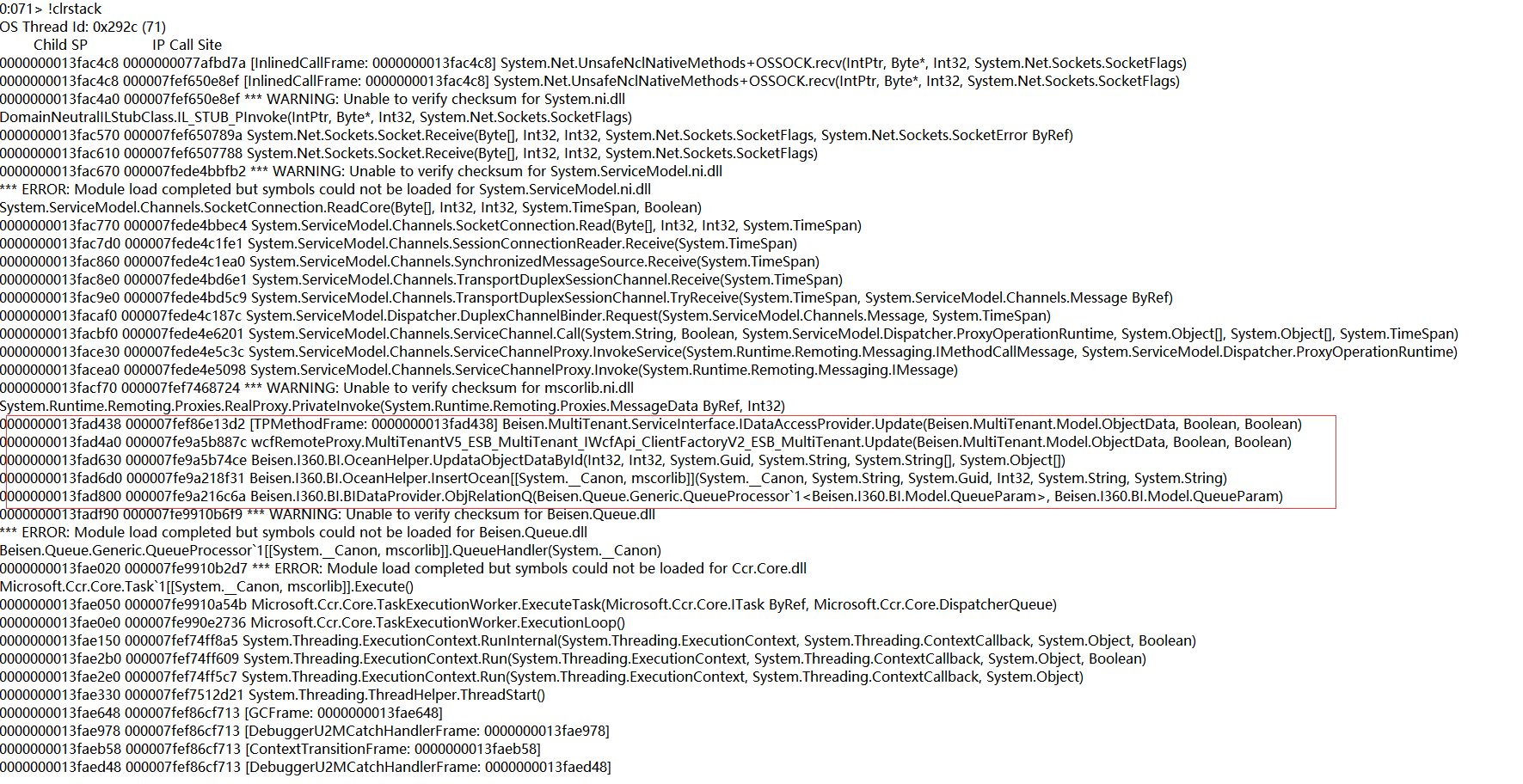
6.在第5步找到第4步查出来的ThreadOBJ，找到线程ID

7.~线程IDs

功能：进入该线程

8.!clrstack

功能：查看该线程在干啥

效果图：

1. **学习资源**
2. [**https://docs.microsoft.com/zh-cn/dotnet/framework/tools/sos-dll-sos-debugging-extension**](https://docs.microsoft.com/zh-cn/dotnet/framework/tools/sos-dll-sos-debugging-extension)
3. [**https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/getting-started-with-windows-debugging**](https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/getting-started-with-windows-debugging)
4. [**http://note.youdao.com/noteshare?id=d285a787b141b4c951c56e35269d22eb**](http://note.youdao.com/noteshare?id=d285a787b141b4c951c56e35269d22eb) **(我自己写的一些)**

!dumpheap –stat //查看所有对象堆栈信息

!dumpheap -mt [000007fef8a3aee0] -min 200000 //查看模块大于200000字节的对象

!do [address] //查看对象信息

!gcroot 0000000012a17048 查看对象引用关系

!runaway //查看所有耗时线程

~107 s //进入线程107

!clrstack //查看堆栈信息