* logging系统使用
* 实时日志工具Sawbucks.

安装下载地址: <http://code.google.com/p/sawbuck/>

如果只是在开发机上调试, 不发布的话, 我们按照shell\_main\_delegate.cc的注释描述直接使用content shell的guid {6A3E50A4-7E15-4099-8413-EC94D8C2A4B6}.

1. 安装Sawbucks
2. 生成Guid. VS->Tool->Create Guid, 定义Guid变量;
3. 注册Guid到LogEventProvider.

logging::LogEventProvider::Initialize(kBOLTBrowserProviderName);

在delegate::PreSandBoxStartUp()或者BasicStartUpComplete中调用都行.

1. 在” HKLM\SOFTWARE\[Wow6432Node\]Google\Sawbuck\Providers”下新建注册表项

{6A3E50A4-7E15-4099-8413-EC94D8C2A4B6}, 添加三个值:

"default\_flags"=dword:00000001

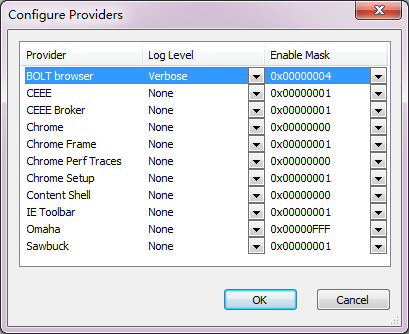
"default\_level"=dword:00000004

默认="Bolt Browser"

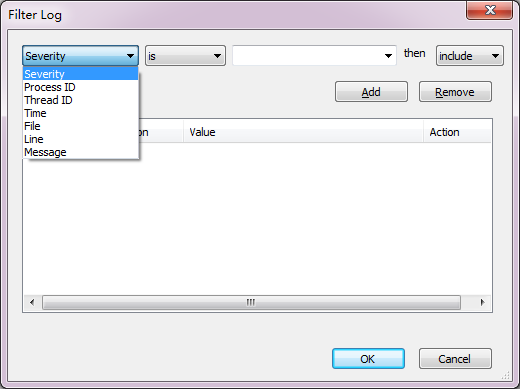
1. 实时监视日志:

Sawbucks -> Log -> Configure Providers 列表中BOLT Browser 项的Log Level置为所需, Enable Mask = 0x00000000 可以显示Log的源文件和行数, Enable Mask = 0xffffffff 源文件路径和行数不可见.

(Todo : Enable Mask还有更细致的定义, 不知道在什么地方设置)



Log -> Filter 制定过滤规则, 去掉太多的冗余信息或者只保留关心日志



Log -> Capture 启动抓取日志功能. 这之后符合条件的浏览器日志都会显示, 但是不能保存. 因为有过滤, 所以可以去掉chrome和content里不需要的信息. 在启动浏览器之前之后都有效.

1. 保存日志和import日志.

按照这个<https://code.google.com/p/sawbuck/wiki/HowToCaptureLogsOffline的方法>, 启动和关闭logman, 注释掉chrome和其他无关的Log session (Kernal也不需要), 加上对Bolt Browser的监听, 要指定刚才生成和注册的GUID:

logman update trace -ets "Log Session" -p {6A3E50A4-7E15-4099-8413-EC94D8C2A4B6} 0x00000000 5

参数0x00000000是Mask, 表示所有日志路径都可见; 最后一个参数5表示抓取日志级别是Verbose (4-Information, 3-Warning, 2-Error, 1-Fatal, 0-None).

执行这两个脚本不需要安装Sawbucks, 可以在测试机上抓取log, 开发加载来看.

在Sawbucks -> File -> Import Log导入执行脚本后生成的文件log.etl.

1. Chrome和Chrome Perf Traces两个provider可以抓取chrome的日志.
2. 还没解决的问题:

Sawbuck下面有一个调用栈, 可以显示Error和Fatal的调用栈, 但是我只弄出来过一次, 就再也没有重现过.

Sawbuck -> Log -> Symbol path 指定符号url, 我想这是本地没有源代码也能打印日志源文件和行数的原因. 但是不知道怎么配置这个.

Mask的细化.

之所以研究这个是因为如果是有些难以重现的问题带着只抓了那么一次的日志包含的内容可能太多了, 又没有机会屏蔽掉冗余信息重现一次, 如果在启动浏览器和关闭浏览器时我们可以使用脚本开启监视, 留存的日志会好看得多. 这个需要启动浏览器时支持.