**Chromium开发的那些事**

Chromium又一次成为公众关注的热点话题，不是因为其市场占有率，而是因为Microsoft Edge，Microsoft的“断腕”举措，使得这个开源项目走到了历史的新高度，毋庸置疑，这是Google的胜利，也是互联网的一件大事情。当一个项目发展到“傲视”这个领域的时候，不知道意味着什么，会不会“盛极而衰”？历史上的一幕一幕会重新上演吗？客观上看，“Chromium”是一个极其优质的开源项目，无论浏览器竞争会怎样结局，Chromium作为一个优秀的软件资产会一直发展下去。

**Chromium再次回到了我的视线**

2009年10月，我第一次下载了最早期的Chromium源代码，记得当时的源代码还需要SVN，一晃十年过去了，一个程序员最宝贵的十年，让我想起很多难忘的往事，那些刻骨铭心的记忆，可能是一生的烙印永远都不会消失……。2017年底，经历了一系列的失败，创业又回到了起点，身边的开发者一个一个的离开了，新的开始会从哪里展开呢？多少年的开发形成的个性，导致我的思维似乎与现在的年轻开发者之间形成了非常大的反差，很多事情让我困惑，选择新的起点真的是一件非常难的事情，这个期间与年轻人谈了很多，一个话题之中谈起了NPM，自然关联起Electron，于是内心里早期的萌动又一次被激活了，我想起了Chromium，历经了多年的微软企业项目开发之后，重新想起了Chromium，一个久违的名字，那一刻，一个新的冲动很快占据了思维的全部，于是我很快决定从Chromium开始，重新建立开发队伍，面向大型开源项目寻找机会的念头变得一发不可收拾……

一切远没有像想象的那样容易，十年时间，早已经物是人非了，Chromium已经是一个庞然大物，绝对不是2009年的那个Chromium了，那一刻，似乎没有回头路了，你说出的话是不能收回的，只能一往无前，所以那一刻就有了新的故事，那些纠结、那些人、那些点点滴滴，每一刻都像是昨天的事情，令人回味无穷。

要打造一个什么样的Chromium版本？两年前的构思其实是非常模糊的，毕竟那个决定以今天看来非常的仓促，模模糊糊的目标在2017年年底不好明确下来，只能摸石头过河，且走且看……。

两年后，结果渐渐明朗化了，我们给出了一个通用的可宿主的Chromium版本，允许一个Window 桌面应用方便的具备“Built-in Browser”，允许开发者有自己的html DOM模型，不同于CEF，我们允许Win32内置全功能的Chromium浏览器，不限于开发者采用的技术必须是C++，开发者可以基于.NET/Java/C++/COM技术构建自己的Browser以及Built-in Browser的应用，例如，开发者可以用.NET控件对象与HTML页面合成新的Web页面，开发者完全可以将自己的技术叠加在Chromium架构之上，形成自己独特的符合云时代应用特征的桌面应用。

我们最终的结果是，将Chromium编译成一个全功能的DLL，形成一个“Chromium Runtime”，这个构思是独一无二的，有这个机制，相当大一类Win32应用直接就是一个定制化版本的浏览器。

现在，即使你仅仅熟悉.NET或者WPF，也可以非常迅速的构建你自己的浏览器了，构建内置浏览器的桌面应用，是一件很Cool的事情，想想看，你的应用会内置一个浏览器，支持由你的组件与传统页面合成的新型页面，你可以给浏览器提供一个基于.NET或者MFC View的侧边栏，这一切是不是很有意思？

**重新组队**

2017年12月中旬，我们开始了Chromium的定制化开发，虽然一直在做微软的企业项目，Google的套路我大致心里还是有数的，Chromium可以算是一个体量很大的项目，不是简单的几台机器就可以玩得转的，对年轻人而言，应该是一个全新的开始，不是一个简单的新项目，需要大内存、高配置的工作环境，这一切，对从来没有经历的人而言，是无法想象的，为了做与Chromium相关的开发，我们一开始就组织了4台128G内存的PC服务器，凭借自己的直觉，大致做了最早期的一些分析，我告诉年轻人，完整的一次编译，大概需要10小时，内存的基本配置不会低于64G，最初的这个判断完全是基于过去的直觉与记忆，身边的人困惑的接受这些安排，因为接下来他们将面对所有的全新的、不可预料的开发过程，需要重新建立信任、重新彼此了解，他们几年来一直与我朝夕相处，面对这样一个决定，一定会是觉得非常的突兀，因为在他们的印象里面，Google的技术似乎与我完全没有关系，同时也不是我知识体系里面的一部分，所以，那一幕徐徐拉开的时候，其中的忐忑可想而知……。

**获取代码、第一次编译Chromium**

获取代码，是我们的第一步。当然需要翻墙，多年的企业开发，让我极其重视硬件部署，队伍里面留下的年轻人都知道这一点，所以，操作服务器、各种虚拟化都不是问题，翻墙，自然是小菜一碟。通过一两天的折腾，我们成功的获取了2017年12月的Chromium源代码，第一次编译我们使用了VSCode，可以说，这个第一次，意义重大，大家第一次看到了python脚本组织的大型C++编译流程，第一次领略了一个需要10小时编译的项目，第一次真实的操作10G体量的C++工程，对绝大多数程序员而言，这些以前只能听别人说的故事真真切切的发生了，第一个自己编译的Chromium浏览器跑起来的时候，那是一种完全不一样的感觉……

**为什么是Chromium？**

为什么选择Chromium做为新起点？直到今天也说不清楚，或者说是冥冥之中的注定，也许是一种必然，谁知道呢？！可能有一个原因是，大家一直觉得我是一个怪人，想法几乎与其他人格格不入，技术侧重点也是，我选择Chromium的一个原因是重新建立大家的开发习惯，让大家真实的体验一下Google人的思维是不是与大众格格不入，因为这类人少，所以很难被大家理解，或许这样就可以理解我的一些所谓的怪异之处。事实上，真实的原因当时很难说清楚，我对浏览器技术有长达二十年的技术积累，或许这才是真实的依据所在。

**下一步是什么？**

Chromium是一个开源项目，其实这个体量的开源项目不折不扣的是一个巨大的陷阱，除了全部的源代码，接下来的茫然，会使得你无所适从，该怎样开始？编译成功了，下一步是什么？读文档？浩如烟海的文档，会将激情淹没的，如果没有办法走出下一步，那么刚刚开始建立起来的一点点兴致就会很快的消失，结果将是放弃，没办法，只能凭着感觉走下去……。

**一切从CreateWindowEx开始……**

当年轻人从Google官方网站获取代码的时候，我就开始构思下一步该怎么开始，我用手里的一台笔记本跑起来Chromium的编译环节，通过一个晚上的分析，设置了大概10个左右的断点，我当时的想法是，必须让代码与实际创建的对象真实的对应起来，所以断点的设置必须精准、与Chromium的流程合理的匹配，这是培养感觉的第一步，非常关键，应该说这是一个至关重要的一刻，让其他人信任你，这一刻是建立信任的开始。

编译成功后不久，我开始与其他成员分析Chromium的运行机制，第一步是在源代码里面搜索“CreateWindowEx”，因为Chromium是一个Win32程序，只要创建窗口对象，就会调用CreateWindowEx这个函数。尽管有10G的源代码，里面包含这个函数的环节的的确确非常的少，以Window对象为线索，入口点瞬间变得非常少了，可选择的地方也变得非常的清晰，听到这些，迷茫逐渐减弱了。

**使用Spy++**

为了具体定位CreateWindowEx，我启动了一个Chrome浏览器，然后打开Visual Studio附带的Spy++，对企业应用开发者而言，很少使用这个工具软件，对C++经验不足的程序员而言同样很少使用。用Spy++分析Chrome的浏览器窗口层次结构，是解开困惑最直接的一个环节，在Spy++打开之后，程序员们清晰的看到如下的场景：

图片包含 屏幕截图

描述已自动生成

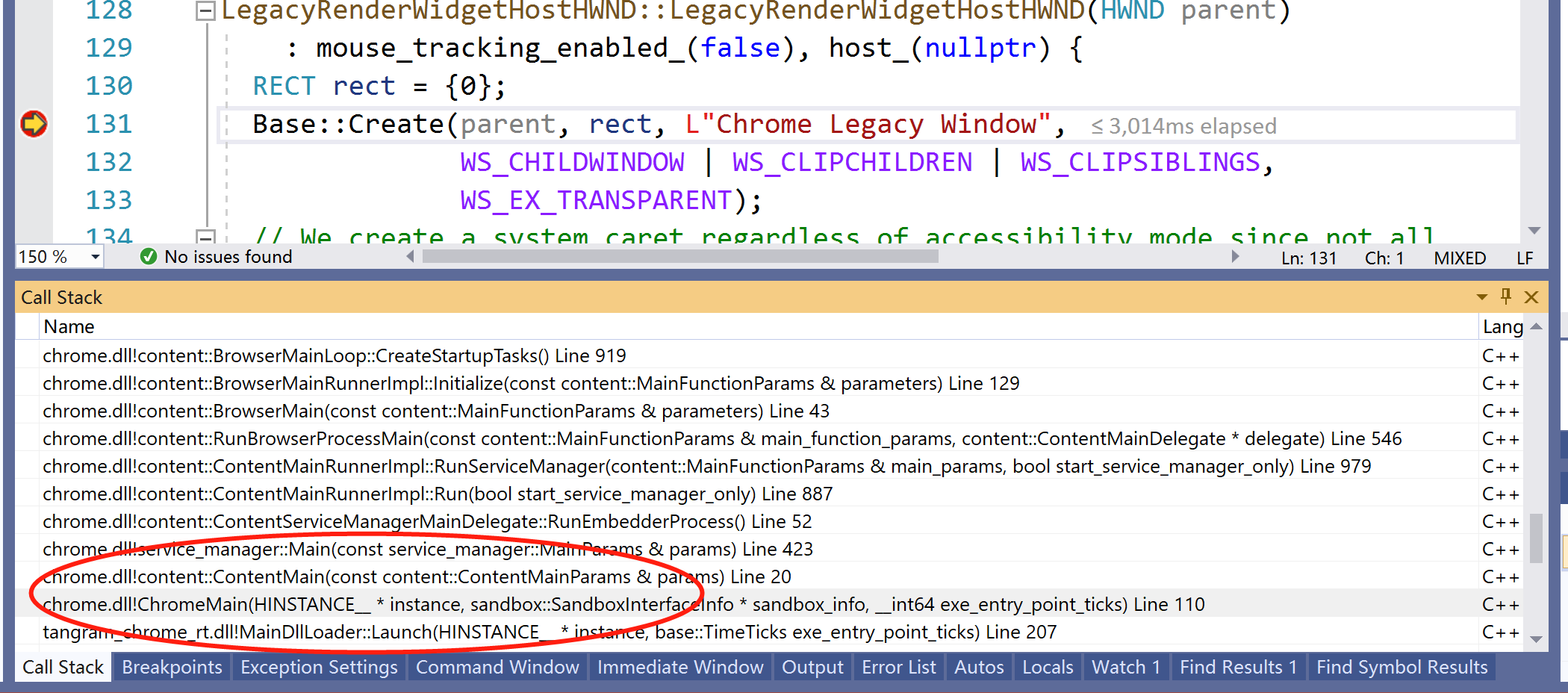
整体上看，Chromium的窗口层次非常简单，我们看到一个父窗口里面包含两个子窗口，父窗口的窗口类是“Chrome\_WidgetWin\_1”，一个子窗口的窗口类是“Chrome\_RenderWidgetHostHWND”，另外一个窗口的窗口类是“Intermediate D3D Window”，这些是Spy++提供的信息。通过进一步利用Spy++，我们发现，两个子窗口分属不同的进程，其中一个与父窗口在同一个进程，另外一个“Intermediate D3D Window”属于另外一个进程，如果我们打开第二个独立的浏览器窗口，我们发现一个新的“Intermediate D3D Window”窗口创建了，与第一个窗口属于同一个进程。以前年轻开发者只是知道Chromium是多进程的，通过以上的分析，的的确确如此。

通过进一步搜索“Chrome\_WidgetWin”以及“Chrome\_RenderWidgetHostHWND”，我们最终锁定了三个文件，分别是“legacy\_render\_widget\_host\_win.h”，“legacy\_render\_widget\_host\_win.cc”以及“window\_impl.cc”，通过Spy++，我们发现窗口类“Chrome\_RenderWidgetHostHWND”对应的窗口标题是“Chrome Legacy Window”，这样，我们很快找到了第一个“断点的位置，如图所示：

图片包含 屏幕截图

描述已自动生成

当调试启动之后，断点会显示如下场景：



顺着调用堆栈的提示，我们会发现一系列函数的名字，通过源代码搜索对应的函数，我们不难发现一组可以设置断点的函数，这样Chromium的代码启动的基本逻辑也就逐步变得清晰了……