

Flash player profiling

Types of profiling

There are 2 types of profiling according to adobe:

Performance profiling is the process of looking for methods in your application that run slowly and can be improved. Once identified, these hotspots can be optimized to speed up execution times so that your application runs faster and responds more quickly to user interaction. You generally look for two things when doing performance profiling: a method that is called only once but takes more time to run than similar methods, or a method that may not take much time to run but is called many times. You use the performance profiling data to identify the methods that you then optimize. You might find that reducing the number of calls to a method is more effective than refactoring the code within the method.

Memory profiling is the process of examining how much memory each object or type of object is using in the application. You use the memory profiling data in several ways: to see if there are objects that are larger than necessary, to see if there are too many objects of a single type, and to identify objects that are not garbage collected (memory leaks). By using the memory profiling data, you can try to reduce the size of objects, reduce the number of objects that are created, or allow objects to be garbage collected by removing references to them.

Profilers

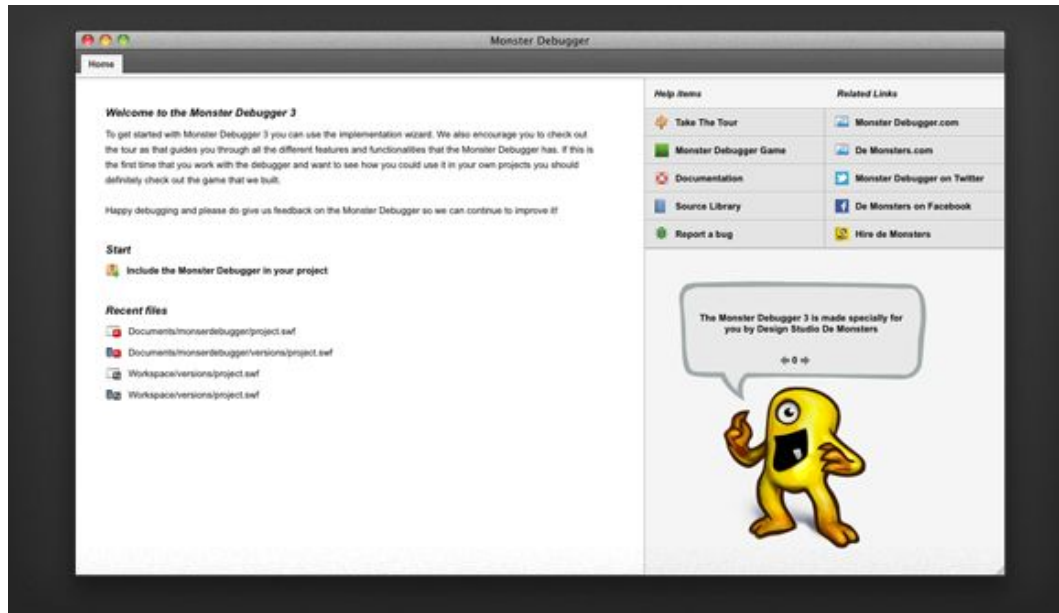
As result of an investigation on flash profiling, I identified 2 types of flash profiling: code intrusive, which consists of adding extra code to the flash application which enables another program to connect and retrieve the information, and non intrusive which does not require extra coding to work.

Intrusive

I managed to test these 2 but not inside our flash code, but in certain examples given in their respective home pages.

Monster debugger

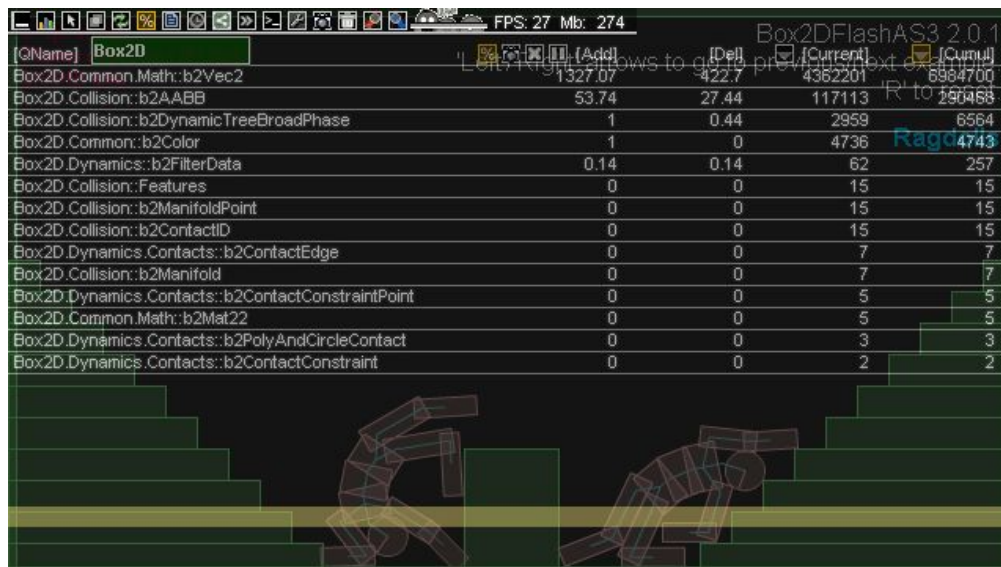
It is an open source Air application to profile flash applications. It also requires to include an swc and instantiate a class in the application to run. It also enables provides a way to debug the flash code of the application running, also providing means to edit the variables in runtime.



The miner profiler

It is a free (for non commercial use) application which enables to profile flash applications. It requires to include a swc in the project and instantiate a class of it to work.

This profiler can also work as “non intrusive” by configuring mm.cfg (explained later) or using flashdevelop to do so.

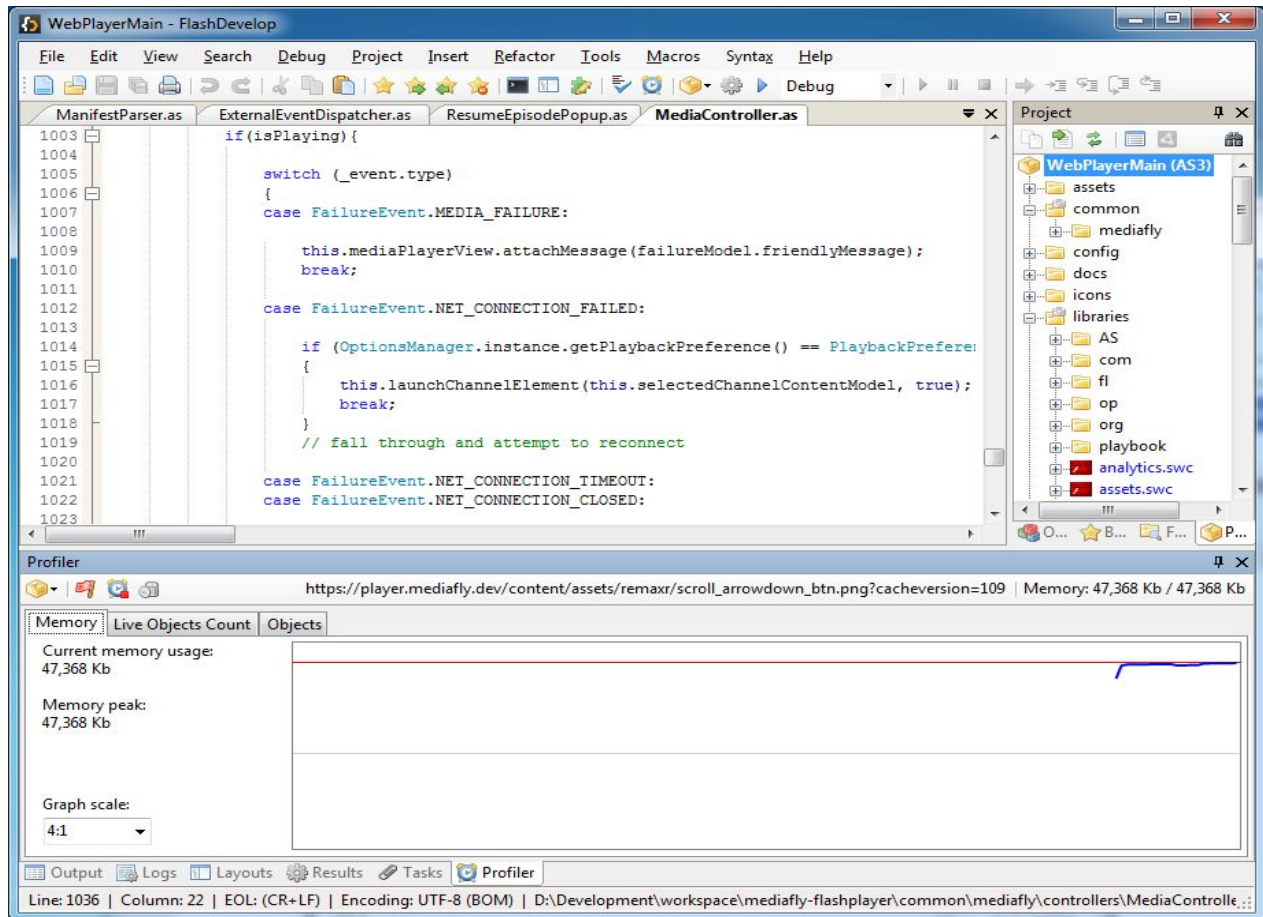


Of this 2 applications the miner looks more profiling oriented, but it is the more intrusive of both applications, not only must you add their code to the application but also the information will shown on top of the application.

Non Intrusive

Flashdevelop

Flashdevelop has its own debugger/profiler, it does not require to include any sort of external code to achieve this.



Flash Builder profiler

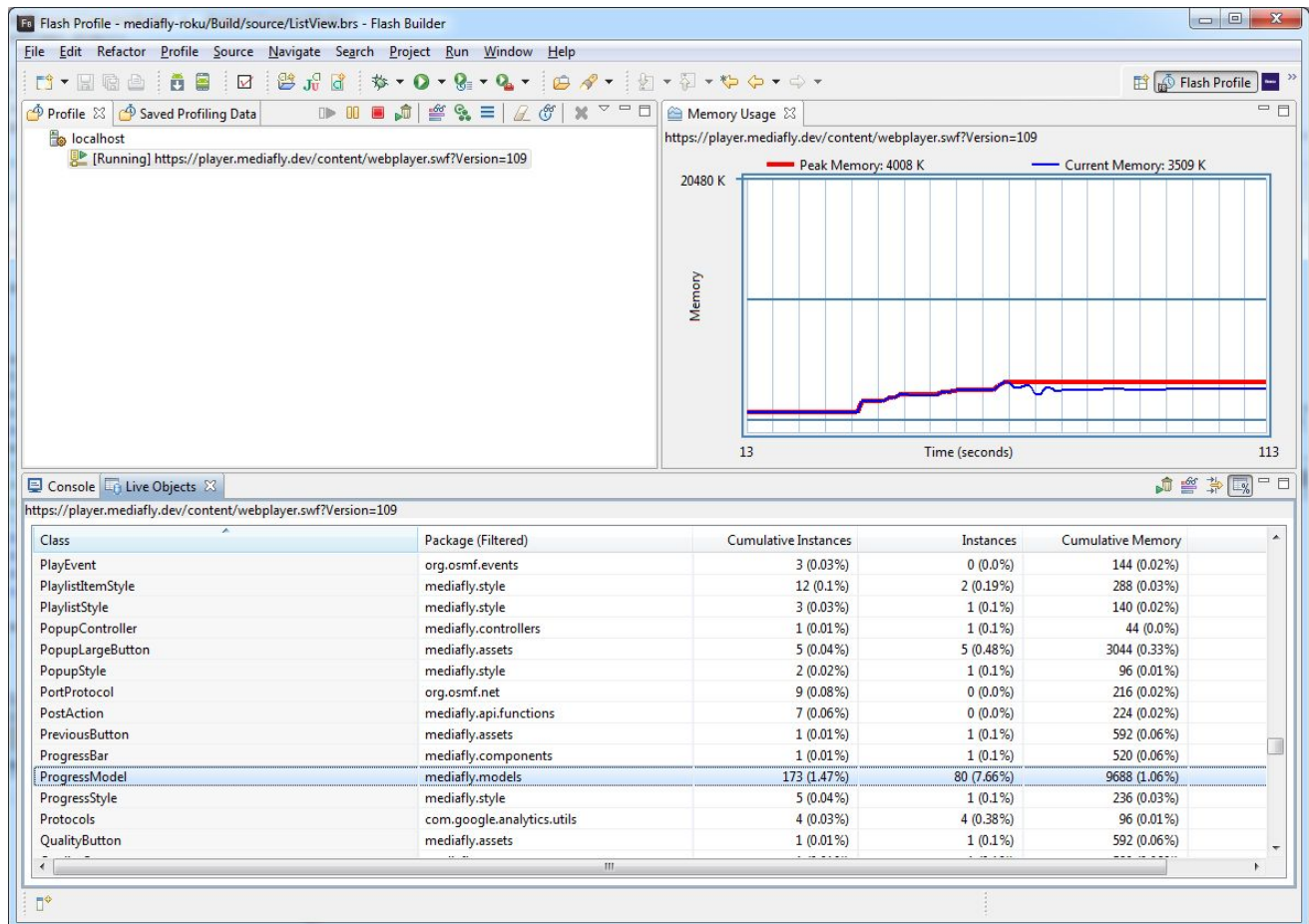
Flash builder has its own flash profiler/debugger, I found it to be the most complete of them all, also adding it has a non intrusive way of profiling the applications and works ok with the applications running in a browser (it seems to have some problems when profiling applications inside firefox, but it works ok with chrome). I had to adapt the project (which is actually made for flashdevelop to work with flash builder) so this could be profiled with this tool.

After many configuration and tries I could have this profiler working, This profiler isn't as stable as it should be, but at least it works, I couldn't determine why the profiler wouldn't work, but I found out the minimal requirements for it to do.

mm.cfg: is the configuration file of the debugger flash player, it should be located under "C:\Users\{User name}" folder, and it must include the parameter "PreloadSwf=D:/Development/workspace/.metadata/.plugins/com.adobe.flash.profiler/11/Profiler Agent.swf?host=192.0.0.1&port=8813" to work. Apparently that PreloadSwf is set by the Flash builder profiler every time it is started, and after the profiler stops working it is deleted.

Firefox: I couldn't have the profiler working with firefox, I believe it has something to do with

flashfirebug, but it worked fine with chrome.



For the profiler to work properly the debugger version of flash must be installed and enabled. After downloading and installing the debugger version one must enter "chrome://plugins/" in chrome's address bar and then disable the default flash player and enable the other.

Flash Builder

I wrote this description of Flash Builder on response to a request Laimonas made after I recommended the tool to him.

Flash builder is an eclipse based flash/flex/as3 editor.

Adobe Flash Builder 4.6 Standard Edition: This is a professional-grade development tool designed to help software developers rapidly build cross-platform RIAs and content using the open source Flex framework. Access web services, REST APIs, and other data sources.
US\$249

Adobe Flash Builder 4.6 Premium Edition: This edition has all the great features that Standard offers plus professional testing tools, including profilers, network monitoring, unit testing support, and command-line build, and includes a free license for Adobe ColdFusion® Builder™ software. US\$699

Moreover, as being an eclipse based tool, there are other (free) eclipse plugins which can also be installed, this way making it possible to have a better and more integrated working environment. I've installed and been working with a mercurial plugin, a plugin to edit roku's brightscript, a plugin to work with python and one to develop applications for android. I also installed an ant plugin which is the one I use to build the roku application packages.

I also managed to export this working environment recently, without having to reinstall all the plugins, which makes it very easy to have it working in case it must be replicated somewhere else.