



Flex Application Profiling by Example

Jun Heider, Sr. Developer / Technical Trainer
@RealEyes Media



Who am I?

- Part of the RealEyes Media team. RealEyes is an Adobe partner with a concentration in delivering Flash Platform solutions.
- A Flex Developer, just like you.
- A Flex Builder debugging and profiling enthusiast.
- Experienced in building many Flex and AIR applications for a number of clients: Abeka Academy, Beatport.com, Chase to name a few.



What this session is not

- An extended lecture based on tons of slides.
- A detailed explanation of memory management and performance theory.
- 100, 300, or 400 level

What this session is

- A quick discussion of things to keep in mind when you profile your applications.
- A whole lot of live Flex Builder Profiler demonstration with actual applications.
- A 200 level presentation.
- As plain English as possible, let's get down to what really matters.
- Hopefully a lot of fun!

Why do we profile our apps?

- Because we want them to run good.
- Because we don't want them to be memory hogs.
- Because we don't want them to crash.

Why do we use the Flex Builder Profiler?

- To find memory leaks.
- To find excessive allocation.
- To find long running processes.
- To find excessively called processes.



Flex Builder Profiler doesn't report everything*

- Doesn't report Browser memory.
- Doesn't report the following OS memory:
 - Memory used to display visuals.
 - Memory used to get events.
 - Memory used to handle Network or File Access.
- Doesn't report the following Player memory:
 - Memory used to track objects. (There are memory control blocks used for every object and hidden objects for all display objects.)
 - Memory used to store player code and pre-defined variables.
 - Memory used by the JIT buffer.
- Memory used by sub-objects or Strings is tracked on its own row.

*Contents of this slide extracted from: http://blogs.adobe.com/aharui/2008/09/using_the_flex_builder_3x_prof.html (embedded in the presentation SWF)



System.totalMemory doesn't report everything either*

- Doesn't report Browser memory.
- Doesn't report the following OS memory:
 - Memory used to display visuals.
 - Memory used to get events.
 - Memory used to handle Network or File Access.
- Doesn't report the following Player memory:
 - Memory used to store player code and pre-defined variables.
 - Memory used by the JIT buffer.

Operating environment can skew results

- Think of these factors when profiling:
 - Different system specs
 - Debug player vs. standard player
 - AIR on different operating systems
 - Different browsers
 - Different versions of the player
 - Debug build vs. release build
 - Different versions of the Flex framework

Just 'cause you read it don't mean it's true

- Different environment = Different results
- Different code = Different results
- One round of testing = Likely anomaly



Verify for yourself!

- Test it with your code base.
- Test it a number of times to verify that the results are consistently worthwhile.
- Make sure to verify that your results will carry over into a release build running in the standard player. (No Profiler but you can observe)

Rules of thumb

- Think about performance from the get-go but don't get carried away.
- Remember that eventually someone will have to maintain the code. (Readability/Clarity)
- Remember that ultimately someone else is usually paying the bill for your optimization efforts.

Rules of thumb (cont.)

- Sometimes the result isn't worth the effort.
- Beware of large late-stage optimization efforts.
- Establish some baseline performance requirements from stakeholders.

What's new in Flex Builder 4 Profiler?

- Improved Object References panel. The shortest paths from the object to the GC root is much easier to find.
- Additional preference configuration options:
 - Configure which back reference paths to view.
 - Configure the maximum number of back reference paths to find.
- Personal observation: The Profiler seems to work much faster and seems to be more stable.

Now for the demos (Time permitting)

- A couple applications that load various data and assets.
- An application implemented in several popular frameworks. (Thanks to Tony Hillerson and InsideRIA FrameworkQuest!)*
- Flex 4 vs. Flex 3 (Hello World application)
- Various algorithm tests.

*You can find the FrameworkQuest here: <http://www.insideria.com/2008/12/frameworkquest-2008-introducti.html>

Where to go from here.

- Open the AIR application on the USB Flash Drive:
 - Source code
 - Links to additional resources
- You can also visit me online or contact me:
 - (w) <http://www.realeyes.com>
 - (b) <http://www.iheartair.com>
 - (t) coderjun
 - (e) jun@realeyes.com
- Mad props to both Alex Harui and Tony Hillerson!