

Honey? Does this format make my data look fat?



Lori MacVittie, 2008-09-07

CNet is reporting that [Google is ditching XML](#) for a faster, more compact alternative known as [ProtocolBuffers](#). I'm going to type this post *really* fast before Don finds out and starts laughing at me because he's always had this thing against XML, claiming it was too bloated and slow.

Apparently Google, the 800-pound gorilla, is on Don's side of this argument, as it [just blogged about its newest](#) creation, ProtocolBuffers.

From CNet's Blog Post

Google thought of using XML as a lingua franca to send messages between its different servers. But XML can be complicated to work with and, more significantly, creates large files that can slow application performance.

I disagree with the statement that it is *XML* that creates large files. No, no it's not. It's *people* that create large files in a data format, and that can happen regardless of whether it's binary or not. If you've ever worked in [digital cartography](#) or drafting, then you know what I'm talking about. [AutoCAD](#) files are *huge*, and they're binary. It's the application and the

people designing the application combined with the amount of data that's being stored or transferred that determines whether a file will end up large or small. While binary is almost always more compact and more efficient than XML, it isn't *always* the case, nor is it inevitable that XML files will end up large and bloated.

Nutrition Facts	
Serving Size 1 File	
Amount Per Serving	
Calories 260	
	% Daily Value
Fat 13g	20%
Saturated Fat 3g + Trans Fat 2g	25%
Cholesterol 30mg	10%
Sodium 660 mg	28%
Carbohydrate 31g	10%
Fiber 0g	0%
Sugars 5g	
Protein 5g	
Vitamin A 4%	Vitamin C 2%

Bad code can be just as inefficient and slow and bloated as inefficient use of a data format. I'm not saying Google's engineers have written bad code or that they are going to write bad code. In fact they probably won't given their track record. But blaming poor performance on a data format is like blaming poor car performance on the car's frame. There's just too many other factors that go into application performance to single out a data format. Network conditions, server load, server platform, coding techniques, etc... can all impact the performance of an application positively and negatively.

While it's certainly likely that Google will see an improvement in performance by moving to its new data exchange format, it's going to be losing at the same time. It's losing the simple integration and interoperability that comes from a standards-based technology like XML. We've been moving away from EAI-like technology that requires coding and development to integrate applications since the advent of SOA, so it's surprising to see such a services-oriented organization like Google move back into the dark ages of integration with this decision. XML became the lingua-franca of integration because it's much easier to integrate into a meta-data driven architecture, which is really one of the foundational pillars of Web 2.0 and SOA.

I will admit that ProtocolBuffers are intriguing and that given the performance needs of an organization like Google it very well may be necessary for it to move away from XML due at least in part to the performance of modern parsers to something more processor efficient, which certainly sounds like ProtocolBuffers. But it's the rare organization that needs *that* kind of speed and, for the most part, XML will continue to suit the majority of folks just fine.



F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com

©2016 F5 Networks, Inc. All rights reserved. F5, F5 Networks, and the F5 logo are trademarks of F5 Networks, Inc. in the U.S. and in certain other countries. Other F5 trademarks are identified at f5.com. Any other products, services, or company names referenced herein may be trademarks of their respective owners with no endorsement or affiliation, express or implied, claimed by F5. CS04-00015 0113