**[翻译]用Page Speed Activity捕获并分析浏览器渲染**

* 原文：[Capturing and analyzing browser paint events using Page Speed Activity](http://code.google.com/speed/articles/browser-paint-events.html)
* 作者：Bryan McQuade, 软件工程师
* 翻译：[ytzong](http://www.99css.com)

### 安装

Page Speed 是一个 Firebug/Firefox 扩展，安装地址：<http://code.google.com/speed/page-speed/download.html>

### 背景：逐步呈现

快速的网页逐步呈现。即随浏览器的加载而逐步显示其内容。一个逐步呈现的页面给浏览者页面在加载的视觉回馈，并尽快给给用户请求信息。[Google](http://code.google.com/speed/page-speed/docs/rendering.html#PutCSSInHead)和[Yahoo](http://developer.yahoo.net/blog/archives/2007/07/high_performanc_4.html) 都建议逐步呈现页面，比如把 CSS 写在页面 head 中。

对多绝大多数页面来说，有一些很好的实践来优化逐步呈现。一个快速的页面应该先给用户呈现可见的内容(译注：浏览器第一屏)，随后呈现视线外的内容(即当前滚动区域外的内容)。一个快速的页面会在加载和渲染重量级的资源(如图片和视频)之前加载和渲染轻量级的资源(如文字)。

一些众所周知的技术会禁止逐步呈现。使用大量表格，甚至用来布局，在一些浏览器中会使逐步呈现失效。应用样式表在页面后方，即便这些样式表在开始的页面加载中使用不到，也会阻止逐步呈现。

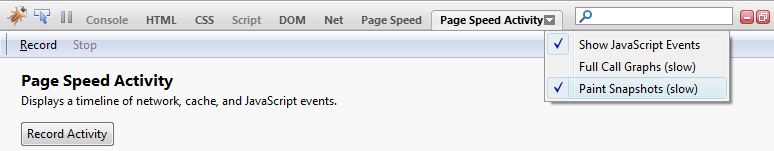
### 用Page Speed Activity捕获浏览器渲染

很难决定一个页面是否要进行逐步呈现优化。大多数页面对肉眼来说呈现太快，以至于意识不到个别的渲染事件(尤其是在高速网络连接下)，而且看不到屏幕区域外的内容是否呈现了。

幸运的是，Firefox 3.5 支持捕获浏览器渲染事件。Page Speed Activity 面板用这个功能可为页面呈现过程录制一个“幻灯片”。幻灯片的每一单元显示了哪一个屏幕区域被渲染(黄色)，哪些区域在屏幕外(灰色)用户看不到。

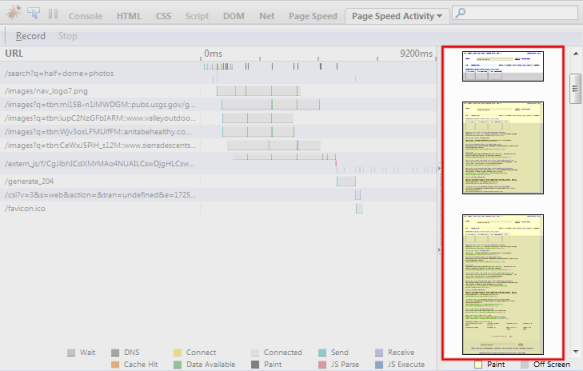
### 启用渲染快照

由于捕获渲染快照会增加一些开销并拖慢浏览器，Page Speed Activity 屏幕快照默认是禁用的。启用渲染快照，确保Activity 下拉菜单中的”Paint Snapshots (slow)” 被选中。



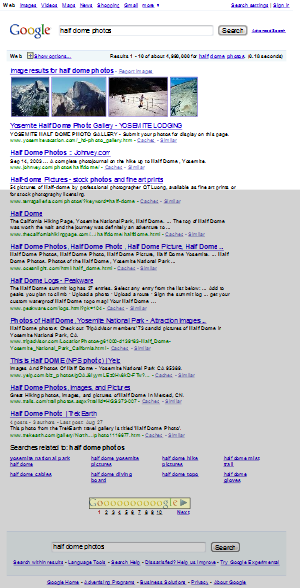
#### Activity 面板中的渲染快照

一旦渲染快照启用，就开始用 Activity 面板录制。Activity 面板将在时间线上显示网络，缓存，和 JavaScript 事件。渲染快照幻灯片出在Activity 面板右侧。渲染快照会按捕获顺序绘制，最早的快照在顶部。拖动右侧的滚动条可看到所有的快照。



### 快照回放示例

译注：请到[这里](http://code.google.com/speed/articles/browser-paint-events.html#animimage)看演示



在这个例子中，我们可以看到以渲染快照慢动作回放的 Google 搜索结果页面的逐步呈现。这些快照用 modem 连接捕获。

用 Page Speed Activity 回放这个例子的渲染快照，请点击 Play Paint Events按钮。

在快照中，用户可见的区域为白色，用户不看见的区域(当前浏览器滚动区域之外)用灰色遮罩。每个快照用黄色遮罩显示重绘区域。

请注意，可见区域的文本内容首先呈现，之后是屏幕外的文本内容。通过先呈现屏幕内的可见区域，用户尽早获取了尽可能多的有用信息。

文本内容呈现后，图像内容接着呈现。推迟图像内容呈现直到文本内容下载完成并呈现完成，这使得浏览器尽早的显示文本内容，再一次尽早给用户尽可能多的有用信息。

在页面标签中给所有图像指定宽度和高度属性，浏览器不需在图像加载后 reflow 页面。尽管没有严格的与逐步呈现有关，指定图像的宽高会带来更好的用户体验，在页面加载时页面内容不会在附近移动。

最后，注意图像自身的逐渐呈现。允许用户在全部图像完成加载前看到图像内容。现代浏览器在 HTML 中用 <img> 标签逐步呈现图像。相反，许多浏览器中用CSS 的background-image 属性不会逐步呈现。为了逐步呈现图像，用 HTML 的 <img> 标签来代替 CSS 的 background-image属性。

### 总结

使用 Page Speed Activity 面板的渲染快照功能，我们可以清晰的观察页面的逐步呈现行为。这使得网页开发者可以为逐步呈现而优化页面，这是快速网页非常重要的特性之一。

### 延伸阅读

* [Page Speed](http://docs.google.com/speed/page-speed/index.html) – 开源的 Firefox/Firebug扩展，评估页面性能并捕获屏幕快照
* [Web Performance Best Practices: Put CSS in the document head](http://docs.google.com/speed/page-speed/docs/rendering.html#PutCSSInHead) – Google 的最佳网页优化实践讨论 CSS 放在页面  
  head 的重要性
* [Response Times: The Three Important Limits](http://www.useit.com/papers/responsetime.html) – Jakob Nielsen 指出了一些关于响应时间的经验准则，并强调当页面加载过程中给用户持续反馈的重要性
* [How Design Affects Performance: Progressive Rendering](http://www.vbulletin.org/forum/showthread.php?t=161099) – 做了一个非常好的并排视觉对比来说明逐步呈现的好处，讨论了HTML表格对逐步呈现的影响
* [High Performance Web Sites: Put Stylesheets at the Top](http://books.google.com/books?id=jRVlgNDOr60C&lpg=PA37&pg=PA37#v=onepage&q=&f=false) – Steve Souders 的第一本网页性能的书介绍了如何加载样式表可以禁止逐步呈现
* [Charles Web Debugging Proxy](http://www.charlesproxy.com) – Charles 允许网页开发者调节他们的网络连接，去模拟低速用户的页面加载