

# ASP.net - How can one differentiate Page-Processing Time from Client-Transmission Time

Asked 16 years, 3 months ago   Modified 16 years, 3 months ago

Viewed 1k times



2



The single timing column in the weblog naturally includes client transmission timing. For anomaly analysis, I want to differentiate pages that took excessive construction time from requests that simply had a slow client.

For buffered pages, I've looked at the ASP.NET page lifecycle model and do not see where I can tap in and codewise measure just the page-processing time before the page is flushed to the client.

I probably should have mentioned that my goal is production monitoring (not test or dev). In addition, the intent is to annotate the weblogs with this measurement for later analysis. Current we liberally annotate the weblogs with `Response.AppendToLog()`. I believe the desire to use `Response.AppendToLog()` somewhat limits my potential logpoints as for instance, the response-object is not viable in `Application_EndRequest`.

Any insight would be appreciated.

[asp.net](#)[page-lifecycle](#)[measurement](#)[transmission](#)[Share](#)[edited Sep 20, 2008 at 15:15](#)[Improve this question](#)[Follow](#)

asked Sep 16, 2008 at 11:15

[cbcclin](#)

381 ● 1 ● 4 ● 8

## 5 Answers

Sorted by:

Highest score (default)



2



You could use a Stopwatch in the BeginRequest and the PreSendRequestContent as mentioned in the other two answers, or you could just use the request's Timestamp in the PreSendRequestContent.

For example, on [SingingEels](#), I added this to the bottom of my Master Page (yes, it's a hack) :

```
<%=DateTime.Now.Subtract(HttpContext.Current.Timestamp).TotalSeconds %>
```

That way I can see how long any page took to actually execute on the server, including hitting the database, etc.

[Share](#) [Improve this answer](#)[answered Sep 16, 2008 at 12:16](#)[Follow](#)[Timothy Khouri](#)

31.8k ● 21 ● 91 ● 129



the easiest way would probably be to use the following events in the global.asax file:

0



```
protected void Application_BeginRequest(Object sender, EventArgs e)
```

```
protected void Application_EndRequest(Object sender, EventArgs e)
```



You could also implement a custom httpmodule

[Share](#) [Improve this answer](#)

[Follow](#)

answered Sep 16, 2008 at 12:04



[Bob Dizzle](#)

1,163 ● 1 ● 8 ● 18

---



0

This depends on the feature set of the performance tools you have. But if you just need to log the processing time then you could follow this approach.



1. Log the starting time in the [HttpApplication.BeginRequest](#) event.



2. Log the elapsed time in the



[HttpApplication.PreSendRequestContent](#) event.

If you just want a specific page then you could check for this in the BeginRequest event. The application events can be attached in Global.asax.

Share Improve this answer

answered Sep 16, 2008 at 12:09

Follow



Kimoz

375 ● 4 ● 11



0

If you want to log on a specific page, I believe asp.net pages' lifecycle begin with PreInit and end with Disposed, so you can log anything you want in those events.



Or, if you want to log on every page, as Bob Dizzle pointed out, you can use the Global.asax file, which has a thousand events to choose from :



<http://msdn.microsoft.com/en-us/library/2027ewzw.aspx>



Share Improve this answer

answered Sep 16, 2008 at 12:11

Follow



thomasb

6,027 ● 10 ● 60 ● 94



0

You could also do your testing right there on the web server. Then ClientTransmission time becomes effectively 0.



Share Improve this answer

answered Sep 16, 2008 at 13:21

Follow



[Joel Coehoorn](#)

415k ● 114 ● 577 ● 813

