ASP.NET Tracing Overview

<http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx>

**.NET Framework 4**

[Other Versions](javascript:;)

http://i3.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

4 out of 8 rated this helpful - [Rate this topic](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#feedback)

ASP.NET tracing enables you to view diagnostic information about a single request for an ASP.NET page. ASP.NET tracing enables you to follow a page's execution path, display diagnostic information at run time, and debug your application. ASP.NET tracing can be integrated with system-level tracing to provide multiple levels of tracing output in distributed and multi-tier applications.

This topic contains:

* [Features](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#Features)
* [Background](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#Background)
* [Code Examples](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#CodeExamples)
* [Class Reference](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#ClassReference)

[Features](javascript:void(0))

ASP.NET tracing offers the following features:

* **Debugging statments**    You can write debug statements in your code without having to remove them from your application when it is deployed to production servers. You can also write variables or structures in a page and trace through the execution path of your page or application.
* **Integrated tracing functionality**   You can route messages emitted by the [System.Diagnostics.Trace](http://msdn.microsoft.com/en-us/library/system.diagnostics.trace(v=vs.100).aspx) class to ASP.NET tracing output, and route messages emitted by ASP.NET tracing to [System.Diagnostics.Trace](http://msdn.microsoft.com/en-us/library/system.diagnostics.trace(v=vs.100).aspx). You can also forward ASP.NET instrumentation events to [System.Diagnostics.Trace](http://msdn.microsoft.com/en-us/library/system.diagnostics.trace(v=vs.100).aspx). For more information, see [Walkthrough: Integrating ASP.NET Tracing with System.Diagnostics Tracing](http://msdn.microsoft.com/en-us/library/b0ectfxd(v=vs.100).aspx).
* **Programmatic access to trace messages**   You can access and manipulate trace messages from your code for finer control over the format of trace messages or for additional processing that you require.
* **Application-level tracing**   The application-level tracing option lets you view the most recent tracing data available without restarting a tracing session and without increasing the amount of tracing data that the server must store. The most recent tracing data is displayed, and older tracing data is discarded.

[Back to top](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#Introduction)

[Background](javascript:void(0))

Tracing appends diagnostic information and custom tracing messages to the output of the page and sends this information to the requesting browser. Optionally, you can view this information from a separate trace viewer (Trace.axd) that displays trace information for every page in an ASP.NET Web application. Tracing information can help you investigate errors or unwanted results while ASP.NET processes a page request.

You can configure individual pages to display trace information. Alternatively, you can configure the application's Web.config file so that all pages display trace information unless the page explicitly disables tracing. Setting application-level tracing is useful because you do not have to change individual pages to enable and disable it.

Trace statements are processed and displayed only when tracing is enabled. You can control whether tracing is displayed to a page, to the trace viewer, or both. For information about how to enable tracing for a page, see [How to: Enable Tracing for an ASP.NET Page](http://msdn.microsoft.com/en-us/library/94c55d08(v=vs.100).aspx). For information about how to enable tracing for an application, see [How to: Enable Tracing for an ASP.NET Application](http://msdn.microsoft.com/en-us/library/0x5wc973(v=vs.100).aspx).

Application-Level ASP.NET Tracing

You enable application-level tracing by using the [trace](http://msdn.microsoft.com/en-us/library/6915t83k(v=vs.100).aspx) element in the Web.config file. When you enable application-level tracing, ASP.NET collects trace information for each request to the application, up to the maximum number of requests you specify. The default number of requests is 10. By default, when the trace viewer reaches its request limit, the application stops storing trace requests. You can configure tracing to store the oldest tracing data (discarding newer items) or the most recent trace information (discarding older items).

|  |
| --- |
| **NoteNote** |
| When you enable tracing for the whole application in the Web.config file, trace information is gathered and processed for each page in that application. To override the application-wide settings, set the **Trace** attribute in that page's [@ Page](http://msdn.microsoft.com/en-us/library/ydy4x04a(v=vs.100).aspx) directive to **false**. Any [Write](http://msdn.microsoft.com/en-us/library/system.web.tracecontext.write(v=vs.100).aspx) or [Warn](http://msdn.microsoft.com/en-us/library/system.web.tracecontext.warn(v=vs.100).aspx) statements that you include in a page's code are stored and returned to the trace viewer only. |

Viewing Trace Information

You can view trace information at the bottom of individual pages. Alternatively, you can use the trace viewer (Trace.axd) to view trace information that is collected and cached by ASP.NET when tracing is enabled. For details about what the trace display includes, see [Reading ASP.NET Trace Information](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#ReadingTraceInformation) later in this topic.

If you want trace information to appear at the end of the page that it is associated with, you can set the **trace** element's [PageOutput](http://msdn.microsoft.com/en-us/library/system.web.configuration.tracesection.pageoutput(v=vs.100).aspx) attribute to **true**. If you enable application-level tracing, but you do not want trace information displayed for some pages, you can set the **Trace** attribute in those pages' [@ Page](http://msdn.microsoft.com/en-us/library/ydy4x04a(v=vs.100).aspx)directive to **false**. For more information about how to configure an ASP.NET application, see [ASP.NET Configuration Overview](http://msdn.microsoft.com/en-us/library/ms178683(v=vs.100).aspx).

By default, application-level tracing can be viewed only on the local Web server computer. To make application-level trace information visible from remote computers, you can set the **trace** element's [LocalOnly](http://msdn.microsoft.com/en-us/library/system.web.configuration.tracesection.localonly(v=vs.100).aspx) attribute to **false**.

|  |
| --- |
| **NoteNote** |
| To help keep the Web application secure, use the remote tracing capability only when you are developing or deploying your application. Make sure that you disable it before you transfer your application to production Web servers. To do this, set the [LocalOnly](http://msdn.microsoft.com/en-us/library/system.web.configuration.tracesection.localonly(v=vs.100).aspx) attribute to **true** in the Web.config file. |

The following example shows an application trace configuration that collects trace information for up to 40 requests. It also enables browsers on computers other than the server to display the trace viewer.

<configuration>

<system.web>

<trace enabled="true" requestLimit="40" localOnly="false" />

</system.web>

</configuration>

Writing Custom ASP.NET Trace Messages

You can append custom trace information to the trace display in an ASP.NET page or to the trace log. The trace information that is written to the trace log is viewable with the trace viewer. For more information, see [How to: View ASP.NET Trace Information with the Trace Viewer](http://msdn.microsoft.com/en-us/library/wwh16c6c(v=vs.100).aspx).

You can write trace information by using the [TraceContext](http://msdn.microsoft.com/en-us/library/system.web.tracecontext(v=vs.100).aspx) class's [Warn](http://msdn.microsoft.com/en-us/library/system.web.tracecontext.warn(v=vs.100).aspx) or [Write](http://msdn.microsoft.com/en-us/library/system.web.tracecontext.write(v=vs.100).aspx) methods. The difference between the two methods is that a message written with the [Warn](http://msdn.microsoft.com/en-us/library/system.web.tracecontext.warn(v=vs.100).aspx) method appears in red text.

The following example shows how to use the [TraceContext](http://msdn.microsoft.com/en-us/library/system.web.tracecontext(v=vs.100).aspx) class to display trace information at the end of an ASP.NET page. A different exception is thrown for each [LinkButton](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.linkbutton(v=vs.100).aspx) control that caused the postback. The error message that is used to initialize the [ArgumentException](http://msdn.microsoft.com/en-us/library/system.argumentexception(v=vs.100).aspx) or [InvalidOperationException](http://msdn.microsoft.com/en-us/library/system.invalidoperationexception(v=vs.100).aspx) instance is displayed in the trace log.

C#

[**VB**](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

<%@ Page Language="C#" Trace="true" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<script runat="server">

void Page\_Load(object sender, EventArgs e)

{

try {

if (IsPostBack)

{

switch (Request.Form["\_\_EVENTTARGET"])

{

case "WarnLink":

throw new ArgumentException("Trace warn.");

break;

case "WriteLink":

throw new InvalidOperationException("Trace write.");

break;

default:

throw new ArgumentException("General exception.");

break;

}

}

}

catch (ArgumentException ae) {

Trace.Warn("Exception Handling", "Warning: Page\_Load.", ae);

}

catch (InvalidOperationException ioe) {

Trace.Write("Exception Handling", "Exception: Page\_Load.", ioe);

}

}

</script>

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="server">

<title>Trace Example</title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:LinkButton id="WriteLink"

runat="server"

text="Generate Trace Write" />

<asp:LinkButton id="WarnLink"

runat="server"

text="Generate Trace Warn" />

</div>

</form>

</body>

</html>

Reading ASP.NET Trace Information

You can view trace information that is appended at the end of an ASP.NET page or in the trace viewer. In both cases, the information displayed is the same. ASP.NET organizes the trace information in a series of tables. For information about how to view trace information in a page, see [How to: Enable Tracing for an ASP.NET Page](http://msdn.microsoft.com/en-us/library/94c55d08(v=vs.100).aspx). For information about how to view trace information in the trace viewer, see [How to: View ASP.NET Trace Information with the Trace Viewer](http://msdn.microsoft.com/en-us/library/wwh16c6c(v=vs.100).aspx).

Trace information appears in the following order.

Request Details

The **Request Details** section displays general information about the current request and response.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Session ID** | The session identification for the specified request. |
| **Time of Request** | The time that the request was made. |
| **Request Encoding** | The character encoding for the request. |
| **Request Type** | The HTTP method (GET or POST). |
| **Status Code** | The status-code value associated with the response. For more information, see RFC 2616 at the [World Wide Web Consortium (W3C) Web site](http://go.microsoft.com/fwlink/?linkid=37125). |
| **Response Encoding** | The character encoding for the response. |

Trace Information

The **Trace Information** section displays the flow of page-level events. If you have created custom trace messages, the messages are displayed in the **Trace Information** section also.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Category** | The custom trace category specified in a [Warn](http://msdn.microsoft.com/en-us/library/system.web.tracecontext.warn(v=vs.100).aspx) or [Write](http://msdn.microsoft.com/en-us/library/system.web.tracecontext.write(v=vs.100).aspx) method call, if any. |
| **Message** | The custom trace message specified in a [Warn](http://msdn.microsoft.com/en-us/library/system.web.tracecontext.warn(v=vs.100).aspx) or [Write](http://msdn.microsoft.com/en-us/library/system.web.tracecontext.write(v=vs.100).aspx) method, if any. |
| **From First(s)** | The elapsed time in seconds since the first trace message was processed. The first trace message appears at the top of the list. |
| **From Last(s)** | The elapsed time in seconds between the processing of the current trace message and the previous trace message. |

Control Tree

The **Control Tree** section displays information about ASP.NET server controls that are created in the page.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Control ID** | The identification for the control. If you have not specified an [ID](http://msdn.microsoft.com/en-us/library/system.web.ui.control.id(v=vs.100).aspx) property for the control, ASP.NET generates an [ID](http://msdn.microsoft.com/en-us/library/system.web.ui.control.id(v=vs.100).aspx) by using the[UniqueID](http://msdn.microsoft.com/en-us/library/system.web.ui.control.uniqueid(v=vs.100).aspx) property. |
| **Type** | The fully qualified type of the control. |
| **Render Size bytes** | The size in bytes of the rendered control (including child controls). This is the size of the actual HTML, XML, or other format that is sent to the browser. |
| **ViewState Size bytes** | The size in bytes of the control's view state (excluding child controls). For more information, see [ASP.NET State Management Overview](http://msdn.microsoft.com/en-us/library/75x4ha6s(v=vs.100).aspx). |
| **ControlState Size bytes** | The size in bytes of the control's control state (excluding child controls). For more information, see [ASP.NET State Management Overview](http://msdn.microsoft.com/en-us/library/75x4ha6s(v=vs.100).aspx). |

Session State

The **Session State** section displays information about values that are stored in Session state, if any. For more information, see [ASP.NET Session State Overview](http://msdn.microsoft.com/en-us/library/ms178581(v=vs.100).aspx).

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Session Key** | The key for data stored in session state, if any. |
| **Type** | The fully qualified type of the object that stores the data. |
| **Value** | A string representation of the data stored in session state, if any. |

Application State

The **Application State** section displays information about values stored in Application state, if any. For more information, see [ASP.NET Application State Overview](http://msdn.microsoft.com/en-us/library/ms178594(v=vs.100).aspx).

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Application Key** | The key for data stored in application state, if any. |
| **Type** | The fully qualified type of the object that stores the data. |
| **Value** | A string representation of the data that is stored in application state, if any. |

Cookies Collection

The **Request Cookies** and **Response Cookies** sections display information about the cookies that are passed between the browser and the server on each request and response. The section displays both persistent and session cookies. ASP.NET creates some cookies automatically, such as those for cookie-based Session state and for forms authentication. For more information, see [ASP.NET Cookies Overview](http://msdn.microsoft.com/en-us/library/ms178194(v=vs.100).aspx).

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Name** | The name of the cookie. |
| **Value** | The value of the cookie, or of subkeys and values if the cookie is multivalued. |
| **Size** | The size in bytes of the cookie. |

Headers Collection

The **Headers Collection** section displays information about request and response message header name/value pairs, which provide information about the message body or requested resource. Header information is used to control how request messages are processed and how response messages are created. For more information about HTTP headers, see RFC 2616 at the [World Wide Web Consortium (W3C) Web site](http://go.microsoft.com/fwlink/?linkid=37125).

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Name** | The name of the header. |
| **Value** | The value of the header. |

Form Collection

The **Form Collection** section displays name/value pairs that show the form element values (control values) that are submitted in a request during a POST operation (postback).

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Name** | The name of the form variable. |
| **Value** | The value of the form variable. |

Querystring Collection

The **Querystring Collection** section shows the values that are passed in the URL. In a URL, query string information is separated from the path information by a question mark (?); multiple query string elements are separated by an ampersand (&). Query string name/value pairs are separated by an equal sign (=). The[QueryString](http://msdn.microsoft.com/en-us/library/system.web.httprequest.querystring(v=vs.100).aspx) property of the [HttpRequest](http://msdn.microsoft.com/en-us/library/system.web.httprequest(v=vs.100).aspx) object returns a [NameValueCollection](http://msdn.microsoft.com/en-us/library/system.collections.specialized.namevaluecollection(v=vs.100).aspx) of query string variables.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Name** | The name of the query string variable. |
| **Value** | The value of the query string variable. |

Server Variables

The **Server Variables** section displays a collection of server-related environment variables and request header information. The [ServerVariables](http://msdn.microsoft.com/en-us/library/system.web.httprequest.servervariables(v=vs.100).aspx) property of the[HttpRequest](http://msdn.microsoft.com/en-us/library/system.web.httprequest(v=vs.100).aspx) object returns a [NameValueCollection](http://msdn.microsoft.com/en-us/library/system.collections.specialized.namevaluecollection(v=vs.100).aspx) of server variables.

|  |  |
| --- | --- |
| **Value** | **Description** |
| **Name** | The name of the server variable. |
| **Value** | The value of the server variable. |

[Back to top](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#Introduction)

ASP.NET Tracing and Diagnostics Tracing

ASP.NET tracing writes messages that are displayed on ASP.NET Web pages and the ASP.NET Trace viewer (Trace.axd). In contrast, the [System.Diagnostics.Trace](http://msdn.microsoft.com/en-us/library/system.diagnostics.trace(v=vs.100).aspx)class is used to trace write messages to the standard .NET Framework trace output (typically a console window). To make it easier to track how your ASP.NET Web pages interact with business objects and other components, you can integrate ASP.NET tracing output with [System.Diagnostics](http://msdn.microsoft.com/en-us/library/system.diagnostics(v=vs.100).aspx) tracing. You can then route all tracing messages to one of these outputs.

A common scenario that uses both ASP.NET tracing and [System.Diagnostics.Trace](http://msdn.microsoft.com/en-us/library/system.diagnostics.trace(v=vs.100).aspx) is Web pages that use middle-tier business objects to interact with data and business rules. You can also use [System.Diagnostics.Trace](http://msdn.microsoft.com/en-us/library/system.diagnostics.trace(v=vs.100).aspx) tracing for pages that use enterprise services such as transactions and queues. In these situations, the business and enterprise components play key parts in the successful execution of the page. In addition, it can help with application analysis to monitor execution flow across the multiple tiers by using a single tracing output. For more information, see [How to: Enable Tracing for an ASP.NET Application](http://msdn.microsoft.com/en-us/library/0x5wc973(v=vs.100).aspx).

Trace Configuration Attributes

The following table shows the attributes that you can use to modify the behavior of application-level tracing in the [trace](http://msdn.microsoft.com/en-us/library/6915t83k(v=vs.100).aspx) element of the Web.config file.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| [Enabled](http://msdn.microsoft.com/en-us/library/system.web.configuration.tracesection.enabled(v=vs.100).aspx) | **true** to enable tracing for the application; otherwise, **false**. The default is **false**. You can override this setting for individual pages by setting the**Trace** attribute in the [@ Page](http://msdn.microsoft.com/en-us/library/ydy4x04a(v=vs.100).aspx) directive of a page to **true** or**false**. |
| [PageOutput](http://msdn.microsoft.com/en-us/library/system.web.configuration.tracesection.pageoutput(v=vs.100).aspx) | **true** to display trace both in pages and in the trace viewer (Trace.axd); otherwise, **false**. The default is **false**.  Note**Note**  Individual pages that have tracing enabled are not affected by this setting. |
| [RequestLimit](http://msdn.microsoft.com/en-us/library/system.web.configuration.tracesection.requestlimit(v=vs.100).aspx) | The number of trace requests to store on the server. The default is 10. |
| [TraceMode](http://msdn.microsoft.com/en-us/library/system.web.configuration.tracesection.tracemode(v=vs.100).aspx) | The order in which trace information is displayed. Set to [SortByTime](http://msdn.microsoft.com/en-us/library/system.web.tracemode(v=vs.100).aspx) to sort by the order in which information was processed. Set to[SortByCategory](http://msdn.microsoft.com/en-us/library/system.web.tracemode(v=vs.100).aspx) to sort alphabetically by user-defined category. The default is [SortByTime](http://msdn.microsoft.com/en-us/library/system.web.tracemode(v=vs.100).aspx). |
| [LocalOnly](http://msdn.microsoft.com/en-us/library/system.web.configuration.tracesection.localonly(v=vs.100).aspx) | **true** to make the trace viewer (Trace.axd) available only on the host Web server; otherwise, **false**. The default is **true**. |
| [MostRecent](http://msdn.microsoft.com/en-us/library/system.web.configuration.tracesection.mostrecent(v=vs.100).aspx) | **true** to display the most recent trace information as tracing output; otherwise, **false**. If this value is **false**, when the **requestLimit** value is exceeded, new requests are not stored. The default is **false**.  Note**Note**  Tracing data that exceeds the limit defined by the **requestLimit** attribute is discarded in favor of the most recent data only when **mostRecent**is **true**. |

[Code Examples](javascript:void(0))

How to and Walkthrough Topics

[How to: Enable Tracing for an ASP.NET Page](http://msdn.microsoft.com/en-us/library/94c55d08(v=vs.100).aspx)

[How to: Enable Tracing for an ASP.NET Application](http://msdn.microsoft.com/en-us/library/0x5wc973(v=vs.100).aspx)

[How to: View ASP.NET Trace Information with the Trace Viewer](http://msdn.microsoft.com/en-us/library/wwh16c6c(v=vs.100).aspx)

[Walkthrough: Integrating ASP.NET Tracing with System.Diagnostics Tracing](http://msdn.microsoft.com/en-us/library/b0ectfxd(v=vs.100).aspx)

[Back to top](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#Introduction)

[Class Reference](javascript:void(0))

|  |  |
| --- | --- |
| Class | Description |
| [System.Diagnostics.Trace](http://msdn.microsoft.com/en-us/library/system.diagnostics.trace(v=vs.100).aspx) | The main class for implementing tracing. |
| [TraceContext](http://msdn.microsoft.com/en-us/library/system.web.tracecontext(v=vs.100).aspx) | Captures and presents execution details about a Web request. |

[Back to top](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#Introduction)

[See Also](javascript:void(0))

Reference

[Back to top](http://msdn.microsoft.com/en-us/library/bb386420(v=vs.100).aspx#Introduction)

Concepts

[ASP.NET Performance, Troubleshooting, and Debugging](http://msdn.microsoft.com/en-us/library/bb398859(v=vs.100).aspx)