Sitecore Online Marketing Suite 1

Analytics Configuration Reference

A Conceptual Overview for Developers and Administrators

Table of Contents

Chapter 1

Introduction

This document provides configuration and usage instructions for the Sitecore Online Marketing Suite (OMS, also called analytics). Sitecore administrators and developers should read this document before implementing OMS.

This document provides an overview of the OMS architecture, details of OMS configuration, pointers to common analytics Application Programming Interfaces, and instructions for troubleshooting OMS.

This document contains the following chapters:

- Chapter 1 — Introduction
- Chapter 2 — Sitecore Analytics Overview
- Chapter 3 — Sitecore Analytics Configuration
- Chapter 4 — The Sitecore Analytics API
- Chapter 5 — Analytics Troubleshooting

Chapter 2

Sitecore Analytics Overview

This chapter describes analytics concepts and features, and then explains the architecture of the Sitecore analytics engine.

This chapter contains the following sections:

- Analytics Features and Concepts
- Multivariate Tests
- Analytics Architecture

2.1 Analytics Features and Concepts

This section describes Sitecore analytics concepts, including instructions to implement the corresponding features.

Important
In this document, references to `web.config` may correspond to entries in `/App_config/include/Sitecore.Analytics.config` on your system. If an entry does not exist in `web.config`, look for a corresponding entry in `/App_config/include/Sitecore.Analytics.config`.

2.1.1 Page Request Tracking

The analytics engine automatically tracks all page requests. You can cancel tracking of specific requests. Most analytics operations involve a sequence of request visited by a Web client. With Sitecore, each page request corresponds to an item in the database. Therefore each analytics operation is associated with a Sitecore item, also known as a page.

2.1.2 Analytics Sessions

An analytics session represents a sequence of HTTP requests from a single Web client in a single browsing session, including any events associated with those pages.

To identify a sequence of HTTP requests from a single user, Sitecore sends an analytics session cookie to the Web client. The name of the analytics session cookie is `SC_ANALYTICS_SESSION_COOKIE`. The value of the analytics session cookie before the first comma character (“,”) matches a value in the `SessionID` column of the `Session` table to identify an analytics session record in the Analytics database. The session cookie expires when the user closes the Web client.

Analytics sessions represent browser sessions. Analytics sessions often, but do not always, correspond to ASP.NET sessions. An analytics session can span multiple ASP.NET sessions. For example, ASP.NET terminates ASP.NET sessions after the number of minutes specified by the `timeout` attribute of the `/configuration/system.web/sessionState` element in the `web.config` file. The first request from a Web client creates an analytics session that corresponds to the ASP.NET session. If the ASP.NET session expires before the next request, ASP.NET creates a new session, but Sitecore does not create a new analytics session. The single analytics session represents the activity of the user within and between ASP.NET sessions.

Global Analytics Sessions

A global session represents multiple individual analytics sessions from a single Web client.

To identify repeat visits from a single user, Sitecore sends a persistent session cookie to the Web client. The name of the persistent session cookie is `SC_ANALYTICS_GLOBAL_COOKIE`. The value of the persistent session cookie matches a value in the `CookieValue` column of the `GlobalSession` table to identify a global session record in the Analytics database. The persistent session cookie expires one year after the last page requested from the solution by the Web client.

2.1.3 Analytics Tags

You can use analytics tags to store analytics attributes about global sessions, each of which represents a different site visitor as identified by a global session. For more information about global sessions, see the section Global Analytics Sessions.

Important
Analytics tags differ from security profiles and analytics profiles.[1] Analytics tags apply to global sessions in individual Web clients rather than to users or individual sessions. For more information about analytics profiles, see the section Analytics Profiles.

Note
Analytics tag values apply to global sessions in individual Web clients. If a user accesses the Web site using two different user agents, the analytics engine records two sessions with separate tag values.

2.1.4 Analytics Profiles

Analytics profiles contain information gathered from each analytics session. You can assign analytics profile values to content items, automatically updating the user's analytics profile when the user accesses that content. For more information about sessions, see the section Analytics Sessions.

Important
Analytics profiles differ from security profiles and analytics tags.[2] Analytics profiles apply to individual sessions in individual Web clients rather than to users or global sessions. For more information about analytics tags, see the section Sitecore.Analytics.AnalyticsGlobalSession.Tags.

Note
Analytics profile values apply to individual sessions in individual Web clients. If a user accesses the Web site using two different user agents, the analytics engine records two sessions with separate profile values.

2.1.5 Analytics Page Events

The Sitecore analytics engine associates zero or more events with each page request. By default, each page request raises the `Page Visited` event. You can configure user actions to raise events, such as when a user clears a form or submits their vote in a poll that uses AJAX instead of ASP.NET postback technology. System operations can generate events, such as when a timer expires or a video reaches its conclusion.

Goals and Failures

Some events can represent specific objectives of the Web site owner, such as if a user accesses a resource or completes a form. Other events may represent failures, such as if a user does not click any of the links in a list of search results. Register specific types of goals and failures using the `System/Analytics/Goal` data template as described in the section How to Register an Analytics Page Event.

Campaign Events

You can associate events with specific campaigns. To register a campaign event, beneath the campaign definition item, insert a campaign event definition item using the `System/Analytics/Campaign Event` data template. You can activate the campaign event using the value of the `sc_camp` query string parameter in the Campaign Link field in the Data section of the campaign event definition item. For more information about campaigns, see the section Campaigns. To change the name of the `sc_camp` query string parameter, see the section The Analytics.CampaignQueryStringKey Setting.

[1] For more information about user profiles, see the Security Reference at http://sdn.sitecore.net/Reference/Sitecore%206/Security%20Reference.aspx.
[2] For more information about user profiles, see the Security Reference at http://sdn.sitecore.net/Reference/Sitecore%206/Security%20Reference.aspx.

How to Register an Analytics Page Event

To register an event:

1. In the **Content Editor**, select the `/Sitecore/System/Settings/Analytics/Page Events` item.
2. In the **Content Editor**, insert an event definition item using the `System/Analytics/Page Event` data template, the `System/Analytics/Campaign Event` data template, or the `System/Analytics/Goal` data template, depending on the type of event.

Note
Code and configuration reference event definition items by name.

Tip
Use patterns to identify events within components, such as `Video Starts` and `Video Concludes`.

3. In the **Content Editor**, in the event definition item, in the **Data** section, enter event parameters.

For instructions to activate event using APIs, see the section How to Report an Analytics Page Event, Goal, or Failure. To activate the event using a URL query string parameter, see the section How to Trigger an Event with a URL Query String Parameter.

Important
You must register an event before using it in any other way.

How to Trigger an Event with a URL Query String Parameter

You can set the `sc_trk` URL query string parameter to the name of an event, goal, or failure. For information about configuring the name of this parameter, see the section The Analytics.EventQueryStringKey Setting.

How to Map Query String Parameters to Events

To configure an arbitrary query string parameter to trigger an event:

1. Add a `<trigger>` element within the `/configuration/sitecore/pipelines/startTracking/processor` element in `web.config` that has a value of `Sitecore.Analytics.Pipelines.StartTracking.ProcessQueryString,Sitecore.Analytics` for the `type` attribute.
2. In the `<trigger>` element, set the `querystring` attribute to the name of the query string parameter.
3. In the `<trigger>` element, set the `eventname` attribute to the name of the event to trigger.

If the URL contains the specified query string parameter, the analytics engine records the specified event.

Note
The default OMS configuration maps the URL query string parameter named `sc_rss` to the event named RSS.

2.1.6 Campaigns

A campaign is a marketing effort to promote awareness of a product, service, or other offering through one or more media channels. You can associate each session with one or more campaigns. OMS users can filter analytics data by campaign.

How to Register a Campaign

To create a campaign:

1. In the **Marketing Center** or the **Content Editor**, select the `/Sitecore/System/Marketing Center/Campaigns` item.
2. In the **Marketing Center** or the **Content Editor**, with the `/Sitecore/System/Marketing Center/Campaigns` item selected, insert a campaign definition item using the `System/Analytics/Campaign` data template.
3. In the **Marketing Center** or the **Content Editor**, in the campaign definition item, in the **Data** section, enter campaign parameters.

2.1.7 Reverse DNS Lookups

A reverse DNS lookup provides information about the owner of an IP address.

2.1.8 GeoIP Lookups

A GeoIP lookup provides information about the owner of an IP address beyond that provided by a reverse DNS lookup. To include GeoIP information in reports, the analytics engine uses an external service that performs lookups.

Note
Contact with a third-party such as MaxMind to obtain GeoIP lookup service.[3] For more information about MaxMind, see the section MaxMind GeoIP Lookup Provider Settings.

2.1.9 Automatic Robot Detection

The Sitecore analytics engine can automatically detect robots such as search engines, classifying their sessions separately from those of actual users. If needed, CMS users can reclassify robot sessions as live sessions.

Sitecore uses a number of criteria to attempt to identify robots. A robot is likely to place a number of requests in a short period of time, is unlikely to support cookies, and is unlikely to request CSS files and media referenced by content.

2.1.10 Analytics Email Distribution

To configure analytics report distribution by email, in the Content Editor, edit the Schedule field in the Data section of the `/Sitecore/System/Settings/Analytics/Reports/Reports` item. If this field contains a value, it must be an XML document with the root element `<schedule>`. Each child of the `<schedule>` element must be a `<report>` element. Each `<report>` element designates the properties of a single email report.

For example, with this XML in the Schedule field:

```
<schedule>
  <report
    reportitem="/sitecore/system/Settings/Analytics/Reports/Reports/Marketing/Metrics"
    recipients="alias@domain.tld"
    recurrence="everyday"
    title="Metrics"
    days="1"
    format="html" />
  <!-- additional <report> elements -->
</schedule>
```

[3] This product includes GeoLite data created by MaxMind (http://www.maxmind.com/). For more information about MaxMind, see http://sdn.sitecore.net/Products/OMS/MaxMind.aspx.

The attributes of the <report> element described in the following table are mandatory:

Attribute	Description
reportitem	The full path of the report definition item.
recipients	Email addresses separated by commas.
Recurrence	Frequency of distribution (everyday, weekdays, Mondays, Tuesdays, Wednesdays, Thursdays, Fridays, Saturdays, Sundays, monthly, or biweekly). Separate multiple values with commas (“,”). This value is not case sensitive.
Title	Email message title.
Days	Number of days of analytics data to include in the report.
Format	Report format (html, pdf, png, rtf, word2007, or excel2007).

The /configuration/sitecore/scheduling/agent element in web.config with type Sitecore.Analytics.Tasks.EmailReportsTask distributes analytics reports by email. This agent runs hourly, but checks and updates the value of the Last Run field in the /Sitecore/System/Settings/Analytics/Reports/Reports item in order to run once each day, shortly after midnight.

Note
Sitecore uses the SMTP server specified by the value attribute of the /configuration/sitecore/settings/setting element in web.config with name MailServer.

Note
You can also distribute analytics reports by email using the Sitecore user interface and APIs.

2.2 Multivariate Tests

You can use the analytics engine to apply multivariate testing (also known as A/B testing). You can use multivariate tests to dynamically set the data source of each presentation control.

2.2.1 Multivariate Test Definitions and Multivariate Test Variables

Each multivariate test consists of a multivariate test definition item containing one or more multivariate test variables. Each multivariate test variable specifies an item to apply as the data source of the presentation control when the system applies that variable. The multivariate test definition item specifies a strategy for the system to select which of the multivariate test variables to apply as the data source for the rendering. Layout details allow the user to select a multivariate test to apply to each presentation control.

How to Create a Multivariate Test

To create a multivariate test:

1. In the **Marketing Center** or the **Content Editor**, select the `/Sitecore/System/Marketing Center/Test Lab` item.
2. In the **Marketing Center** or the **Content Editor**, beneath the `/Sitecore/System/Marketing Center/Test Lab` item, insert a multivariate test definition item using the `System/Analytics/Multivariate Test Variable` data template.
3. In the **Marketing Center** or the **Content Editor**, in the multivariate test definition item, in the **Data** section, in the **Title** field, enter the title of the multivariate test as it should appear in Sitecore user interfaces.
4. In the **Marketing Center** or the **Content Editor**, in the multivariate test definition item, in the **Data** section, in the **Description** field, enter a description of the multivariate test.
5. In the **Marketing Center** or the **Content Editor**, in the multivariate test definition item, in the **Data** section, in the **Test Strategy** field, select a test strategy. For more information about multivariate test strategies, see the section Multivariate Test Strategies,

How to Add a Multivariate Test Variable to a Multivariate Test

To add a multivariate test variable to a multivariate test:

1. In the **Marketing Center** or the **Content Editor**, select a multivariate test definition item.
2. In the **Marketing Center** or the **Content Editor**, beneath the multivariate test definition item, insert a multivariate test variable definition item using the `System/Analytics/Multivariate Test Variable` data template.
3. In the **Marketing Center** or the **Content Editor**, in the multivariate test variable definition item, in the **Data** section, in the **Name** field, enter the name of the multivariate test variable.
4. In the **Marketing Center** or the **Content Editor**, in the multivariate test variable definition item, in the **Data** section, in the **Data Source** field, select the item to apply as the data source for the presentation control when the strategy selects this multivariate test variable.

How to Implement a Multivariate Test

To implement a multivariate test, in layout details, in the properties for the presentation component, select the multivariate test.[4]

[4] For more information about presentation component properties, see http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20Reference.aspx and http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20Cookbook.aspx.

2.2.2 Multivariate Test Strategies

Sitecore provides two multivariate test strategies, and you can implement custom multivariate test strategies. The Random multivariate test strategy selects multivariate test variables randomly each time. On its first use, the Sticky multivariate test strategy applies a random multivariate test variable, and then maintains that data source if the user requests the same page again.

How to Implement a Multivariate Test Strategy

To implement a multivariate test strategy:

1. In the Visual Studio project, create a multivariate test strategy class by implementing the `Sitecore.Analytics.Rules.Conditions.IMultivariateTestStrategy` interface.
2. In the class, implement the `GetTestVariableItem()` method. The first parameter to the `GetTestVariableItem()` method is the context item. The second parameter to `GetTestVariableItem()` is the multivariate test definition item. Implement the strategy to return the appropriate multivariate test variable definition item from the children of the multivariate test definition item. For example, to always apply the first multivariate test variable:

```
public Sitecore.Data.Items.Item GetTestVariableItem(
  Sitecore.Data.Items.Item item,
  Sitecore.Data.Items.Item multiVariateTest)
{
  Sitecore.Diagnostics.Assert.ArgumentNotNull(item, "item");
  Sitecore.Diagnostics.Assert. ArgumentNotNull(multiVariateTest, "multiVariateTest");

  if (!multiVariateTest.HasChildren)
  {
    return null;
  }

  return multiVariateTest.Children[0];
}
```

3. In the **Content Editor**, select the `/Sitecore/System/Settings/Analytics/Multivariate Test Strategies` item.
4. In the **Content Editor**, with the `/Sitecore/System/Settings/Analytics/Multivariate Test Strategies` item selected, insert a multivariate test strategy definition item using the `System/Analytics/Multivariate Test Strategy` data template.
5. In the **Content Editor**, in the multivariate test strategy definition item, in the **Test Variable Strategy Script** section, in the **Type** field, enter the .NET type signature of the multivariate test strategy class.
6. In one or more multivariate test definitions, select the multivariate s.NET type signature of the multivariate test strategy class in the **Type** field of a multivariate test definition item as described in the section How to Create a Multivariate Test.

2.3 Analytics Architecture

The Sitecore analytics engine adds analytics processing logic to each page request.

2.3.1 The Analytics Page Event Queue

Each page request can generate a number of analytics page events. Each analytics page event involves at least one database operation. To avoid any performance impact while rendering pages, the Sitecore analytics engine writes database operations to an event queue in memory. A separate thread invokes the appropriate database operations to remove entries from the event queue when hardware resources are available.

The event queue cannot exceed a configurable number of entries. When requests trigger events that cannot fit in the queue, Sitecore discards new events, storing no record of those events. For instructions to configure the number of entries allowed in the event queue, see the section The Analytics.MaxQueueSize Setting.

Important
Configure the queue size to prevent overwhelming the solution infrastructure that lacks capacity at peak periods, or to reduce the harm of aggressive, unrecognized search engines.

Chapter 3

Sitecore Analytics Configuration

This chapter provides instructions to configure the Sitecore analytics engine, including databases, settings, and providers.

This chapter contains the following sections:

- The Analytics Database
- The Analytics Client Roles
- The Analytics Settings
- The Analytics Provider
- The GeoIP Lookup Provider
- The Site Definition enableAnalytics

3.1 The Analytics Database

The `connectionString` attribute of the `/connectionStrings/add` element in `/App_Config/ConnectionStrings.config` with `name analytics` specifies the database containing analytics data. Multiple Sitecore instances can access the same analytics database. For example, multiple content delivery instances behind a load balancer might populate an analytics database accessed by a separate content management server to generate reports.

Warning
Sitecore supports Microsoft SQL Server for analytics features.

3.2 The Analytics Client Roles

The Sitecore analytics roles control aspects to OMS features in the Sitecore user interface.[5]

[5] For more information about the analytics roles, see the Client Configuration Cookbook at http://sdn.sitecore.net/Reference/Sitecore%206/Client%20Configuration%20Cookbook.aspx.

3.3 The Analytics Settings

Comments above each `/configuration/sitecore/settings/setting` element in `web.config` provide possible values for each setting.

Important
For an important note regarding `web.config`, see the section Analytics Features and Concepts.

3.3.1 The Analytics.AutoDetectBots Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.AutoDetectBots` specifies whether the analytics engine attempts to classify search engines and other Web robots automatically. For more information about automatic robot detection, see the section Automatic Robot Detection.

3.3.2 The Analytics.CampaignQueryStringKey Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.CampaignQueryStringKey` specifies the name of the URL query string parameter used by the analytics engine to trigger a campaign. For more information about campaigns, see the section Campaigns.

3.3.3 The Analytics.EMailFromAddress Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics. EMailFromAddress` specifies the email address to use as the sender for messages generated by the analytics engine. For more information about analytics email distribution, see the section Analytics Email Distribution.

3.3.4 The Analytics.Enabled Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.Enabled` controls whether Sitecore stores analytics information about incoming HTTP requests. To improve performance in environments that do not use analytics, set `Analytics.Enabled` to `false`.

3.3.5 The Analytics.EventQueryStringKey Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.EventQueryStringKey` specifies the name of the URL query string parameter used by the analytics engine to trigger an event. For more information about events, see the section Analytics Page Events.

3.3.6 The Analytics.LogLevel Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.LogLevel` specifies the threshold for logging activity. By default, the analytics engine logs only errors and exceptions.

3.3.7 The Analytics.LongRunningRequest Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.LongRunningRequest` specifies the number of milliseconds

to allow for analytics operations. If an analytics operation exceeds this threshold, that operation will appear as an error in the Sitecore log and analytics reports.

3.3.8 The Analytics.MaxQueueSize Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.MaxQueueSize` controls the maximum size of the event queue. For more information about the event queue, see the section The Analytics Page Event Queue.

Warning
The size of the analytics page event queue cannot exceed memory available to ASP.NET. Reaching the analytics page event queue limit does not always indicate an inadequate queue size. Reaching the queue limit may indicate inadequate hardware to service peak load, an aggressive robot, or other factors.

3.3.9 The Analytics.PerformLookup Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.PerformLookup` specifies whether the instance performs reverse DNS and GeoIP lookups. Only one of the Sitecore instances connected to an analytics database should perform lookups.

Important
Most GeoIP lookup providers, including the MaxMind provider, require Internet access to perform GeoIP lookups. Set `Analytics.PerformLookup` to `false` in environments without Internet access.

3.3.10 The Analytics.PerformLookup.Interval Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.PerformLookup.Interval` controls the interval between lookup operations.

3.3.11 The Analytics.PerformLookup.MachineName Setting

If the `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.PerformLookup.MachineName` is not an empty string and does not match `System.Environment.MachineName`, then Sitecore does not perform reverse DNS or GeoIP lookups on the machine even when the `Analytics.PerformLookup` setting is `true`. This allows a single configuration for all machines in a content delivery environment behind a load balancer not configured for server affinity without allowing multiple content delivery servers to perform redundant queries.

Warning
If a load balancer not configured for server affinity (also known as sticky sessions) distributes a series of requests to different content delivery servers, the lookup operation triggered by the first request may not be complete before Sitecore processes a second request, triggering a redundant lookup. Some GeoIP lookup providers, including MaxMind, charge a fee for each lookup operation. To avoid redundant GeoIP lookups in load-balanced environments not configured for server affinity, configure the `Analytics.PerformLookup` and `Analytics.PerformLookup.MachineName` settings such that a single content delivery instance performs reverse DNS and GeoIP lookups. If the content management instance has access to the Internet, configure only the content management instance to perform lookups.

3.3.12 The Analytics.ReverseDnsLookupTimeout Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `Analytics.ReverseDnsLookupTimeout` specifies a number of milliseconds to allow for reverse DNS lookup operations.

3.4 The Analytics Provider

To update the default analytics provider to store analytics data in a system other than the default Sitecore analytics database, implement a custom analytics provider class that provides the same signature as the default analytics provider. Then, either update the `type` attribute of the `/configuration/sitecore/analyticsManager/providers/add` element in `web.config` with `name default` to reference your class, or add an additional `<add>` element within the `<providers>` element, give it a unique `name` and update the `type` attribute to reference your provider, and set the `defaultProvider` attribute of the `<analyticsManager>` element to that `name`.

Important
Your analytics provider must provide the same signature as the default analytics provider.

3.5 The GeoIP Lookup Provider

The `type` attribute of the `/configuration/sitecore/lookupManager/providers/add` element in `web.config` with `name default` specifies the GeoIP lookup provider.

To configure the MaxMind GeoIP lookup provider, see the section MaxMind GeoIP Lookup Provider Settings.

3.5.1 How to Implement a GeoIP Lookup Provider

To implement a custom GeoIP lookup provider:

1. In the Visual Studio Web application project, create a class that inherits from the `Sitecore.Analytics.Lookups.LookupProviderBase` class.
2. In the new class, implement the `GetInformationByIp()` method.
3. In the `/configuration/sitecore/lookupManager/providers/add` element in the `web.config` file, update the `type` attribute to reference the new class.

3.5.2 MaxMind GeoIP Lookup Provider Settings

To use the MaxMind GeoIP lookup provider, configure the settings described in this section, and then set the `type` attribute of the `/configuration/sitecore/lookupManager/providers/add` element in `web.config` with `name default` to the following value:[6]

```
Sitecore.Analytics.Lookups.MaxMindProvider, Sitecore.Analytics
```

While the OMS comes with a number of free lookups for testing and implementation convenience to ensure that you can utilize OMS' capabilities from day one, the free batch of lookups will expire and more must be purchased from MaxMind.

For information on signing up with MaxMind and purchasing more lookups, refer to the following page: http://sdn.sitecore.net/Products/OMS/MaxMind.aspx.

Important
If you use the MaxMind GeoIP lookup service, you must set the `MaxMind.Url` setting to the URL provided to you by MaxMind. For more information about the `MaxMind.Url` setting, see the section The MaxMind.Url Setting.

The MaxMind.Encoding Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name MaxMind.Encoding` controls the page encoding used by the MaxMind GeoIP lookup provider.

The MaxMind.Format Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name MaxMind.Format` controls the format used by the MaxMind GeoIP lookup provider.

The MaxMind.SecurityToken Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name MaxMind.SecurityToken` controls the security used by the MaxMind GeoIP lookup provider.

[6] This product includes GeoLite data created by MaxMind, available from http://www.maxmind.com/.

The MaxMind.Url Setting

The `value` attribute of the `/configuration/sitecore/settings/setting` element in `web.config` with `name` `MaxMind.Url` controls the URL used by the MaxMind GeoIP lookup provider.

3.6 The Site Definition enableAnalytics Attribute

To enable or disable capturing of analytics data for each Web site, you can set the `enableAnalytics` attribute of the appropriate `/configuration/sitecore/sites/site` element in `web.config` to `true` or `false`.

Tip
If you separate content management from content delivery, to prevent capturing analytics in the content management environment, in the content management environment, set the `enableAnalytics` attribute to `false` for all logical sites.

Chapter 4

The Sitecore Analytics API

This chapter describes the Sitecore analytics Application Programming Interface (API).

This chapter contains the following sections:

- Analytics API Overview

4.1 Analytics API Overview

To use the Sitecore analytics APIs, in the Visual Studio project, add a reference to the `Sitecore.Analytics.dll` assembly in the `/bin` folder within the document root of the Sitecore solution Web site.

Warning
In the Visual Studio project, set the Copy Local property of all references in the project to assemblies in the `/bin` folder to `False`. If you do not set the Copy Local property of a referenced assembly to `False`, Visual Studio can delete assemblies from the `/bin` folder, causing Sitecore to fail.

4.1.1 Sitecore.Analytics.AnalyticsPageEvent

The `Sitecore.Analytics.AnalyticsPageEvent` class represents an event, goal, or failure associated within an analytics page. The `Sitecore.Analytics.AnalyticsPageEvent.Name` property must match the name of one of the registered events. Remaining properties of the event object affect analytics reports. For more information about events, see the section Analytics Page Events.

4.1.2 Sitecore.Analytics.AnalyticsTracker

The `Sitecore.Analytics.AnalyticsTracker` class provides access to the analytics context. The `Sitecore.Analytics.AnalyticsTracker.IsActive` property indicates whether analytics are enabled for the current request.

4.1.3 Sitecore.Analytics.AnalyticsPage

The `Sitecore.Analytics.AnalyticsPage` class represents a page in a session. You can associate events with the previous, current, and next pages in the user's session. For more information about analytics pages, see the section Page Request Tracking.

How to Report an Analytics Page Event, Goal, or Failure

You use the same API to register goals, failures, and generic events. For more information about analytics page events, goals, and failures, see the section Analytics Page Events.

To report an analytics page event, goal, or failure:

1. Check the `Sitecore.Analytics.AnalyticsTracker.IsActive` property to confirm that the analytics engine is active:

```
if (!Sitecore.Analytics.AnalyticsTracker.IsActive)
{
  return;
}
```

2. Confirm that you can access the analytics tracker and the current analytics page:

```
Sitecore.Analytics.AnalyticsTracker analyticsTracker =
  Sitecore.Analytics.AnalyticsTracker.Current;

if (analyticsTracker == null
  || analyticsTracker.CurrentPage == null)
{
  return;
}
```

3. Create a `Sitecore.Analytics.AnalyticsPageEvent` object to represent the event, and set the Name property to the name of the event definition item:

```
Sitecore.Analytics.AnalyticsPageEvent pageEvent =
```

```
  new Sitecore.Analytics.AnalyticsPageEvent()
  {
    Name = "Event Name",
    Key = "Event Key",
    Text = "Event Text",
    Data = "Event Data",
    Integer = 999,
  };
```

Important
The `Sitecore.Analytics.AnalyticsPageEvent.Name` property must match the name of one of the registered analytics page events. For instructions to register an event, see the section How to Register an Analytics Page Event.

4. Pass the event to the `Sitecore.Analytics.AnalyticsTracker.TriggerEvent()` method:

```
analyticsTracker.CurrentPage.TriggerEvent(pageEvent);
```

5. Commit the event:

```
analyticsTracker.Submit();
```

4.1.4 Sitecore.Analytics.AnalyticsPage Extension Classes

The static classes in the `Sitecore.Analytics.Extensions.AnalyticsPageExtensions` namespace provide extension methods for the `Sitecore.Analytics.AnalyticsPage` class to achieve specific objectives.

How to Associate the Search Action with the Previous Page

With a classic Web analytics solution, a search operation typically results in records for two or three page requests: the current page, the search page (if the user had to access a search page to enter a search query), and the search results page. With the Sitecore analytics engine, you can cancel tracking of the search results page, and associate the search event with the page from which the user submitted their query.

To associate the search action with the previous page, and cancel tracking of the search results page:

```
using Sitecore.Analytics.Extensions.AnalyticsPageExtensions;

if (!Sitecore.Analytics.AnalyticsTracker.IsActive)
{
  return;
}

Sitecore.Analytics.AnalyticsTracker analyticsTracker =
  Sitecore.Analytics.AnalyticsTracker.Current;

if (analyticsTracker == null
  || analyticsTracker.PreviousPage == null
  || analyticsTracker.CurrentPage == null)
{
  return;
}

analyticsTracker.PreviousPage.Search(
  searchCriteria,
  hits);
analyticsTracker.Cancel();
```

The call to the `Search()` method generates associates the search event . The parameters indicate the search term and the number of results.

The call to the `Cancel()` method cancels tracking of the current (search results) page request.

Note
Visual Studio 2005 does not support extension methods. If you use Visual Studio 2005, instead of using extension methods, call the corresponding static methods directly:

```
Sitecore.Analytics.Extensions.AnalyticsPageExtensions.Searches.Search(
  analyticsTracker.PreviousPage,
  searchCriteria,
  hits);
```

4.1.5 The VisitorIdentification Web control

The VisitorIdentification Web control helps the Sitecore analytics engine identify robots. The VisitorIdentification Web control generates an HTML `<link>` element that references an empty CSS resource. The Sitecore analytics engine uses this request to help differentiate robots from actual users. For more information about automatic robot detection, see the section Automatic Robot Detection.

Important
To assist the analytics engine in robot identification, include the VisitorIdentification Web control in all of your layouts. For example:

```
<sc:VisitorIdentification runat="server" />
```

4.1.6 Sitecore.Analytics.AnalyticsGlobalSession.Tags

You can use the `Sitecore.Analytics.AnalyticsTracker.CurrentGlobalSession.Tags` collection to access analytics tags. For example, to set the analytics tag named `TagName` to the value `Tag Value` for the current user:

```
if (!Sitecore.Analytics.AnalyticsTracker.IsActive)
{
  return;
}

Sitecore.Analytics.AnalyticsTracker analyticsTracker =
  Sitecore.Analytics.AnalyticsTracker.Current;

if (analyticsTracker == null || analyticsTracker.CurrentGlobalSession == null)
{
  return;
}

analyticsTracker.CurrentGlobalSession.Tags["TagName"] = "Tag Value";
```

Note
You do not need to commit after updating a tag.

Note
You cannot remove an analytics tag using the API.

Chapter 5

Analytics Troubleshooting

This chapter provides instructions to troubleshoot issues with the Sitecore analytics engine.

This chapter contains the following sections:

- OMS Troubleshooting Overview

5.1 OMS Troubleshooting Overview

Use the information in the following sections to troubleshoot issues with the Sitecore analytics engine.

5.1.1 Sitecore Logs

Always begin troubleshooting by investigating the Sitecore log files in the directory specified by the `value` attribute `/configuration/sitecore/settings/setting` element in `web.config` with `name LogFolder`.

5.1.2 Analytics Control Panel

The **Analytics Control Panel** in the Sitecore desktop provides information about reverse DNS and GeoIP lookup success and failure.

To determine if Sitecore is able to perform reverse DNS and GeoIP lookups:

1. In the Sitecore Desktop, click the **Sitecore** button. The **Sitecore** menu appears.
2. In the **Sitecore** menu, click Control Panel. The **Control Panel** appears within the Sitecore Desktop.
3. In the Sitecore Desktop, in the **Control Panel**, click **Analytics**. The **Analytics Control Panel** appears in the Sitecore Desktop.
4. In the **Analytics Control Panel**, click **Run Lookups** and complete the wizard that appears.
5. IIn the **Analytics Control Panel**, click **Retrieve Status Information**. The **Analytics Status** report appears in the Sitecore Desktop.
6. In the **Analytics Status** report, if **Last Successful** for **Requests** does not indicate a recent date and time, or if **Last Failed** for **Requests** indicates a recent date and time, then the system exceeded the event queue. For more information about the event queue, see the section The Analytics Page Event Queue.
7. In the **Analytics Status** report, if **Last Successful** for **Lookups** does not indicate a recent date and time, or if **Last Failed** for **Lookups** indicates a recent date and time, then check the GeoIP lookup provider configuration and confirm that the system has Internet access.

If a reverse DNS lookup or GeoIP lookup operation times out or otherwise fails, information obtained from the reverse DNS lookup provider does not appear in analytics reports. Check the configuration of the GeoIP lookup provider, confirm that the system has Internet access, and investigate the Sitecore log for more information.