



Sitecore 6.5 or later

# Engagement Analytics API Reference Guide

*A developer's reference guide to the Engagement Analytics API*

## Table of Contents

Chapter 1	Introduction.....	3
Chapter 2	The Analytics API.....	4
2.1	Sitecore.Analytics.Tracker .....	5
2.2	Sitecore.Analytics.Data.DataAccess.Visitor.....	8
2.3	Sitecore.Analytics.Visit.....	9
2.4	Sitecore.Analytics.TrackerDataContext .....	10
2.5	Sitecore.Analytics.Data.TrackerChanges .....	12
2.6	Sitecore.Analytics.Data.DataAccess.DataAdapters.SqlBase .....	13
2.7	Sitecore.Analytics.Data.DataAccess.DataSets.VisitorsRow.....	15
2.8	Sitecore.Analytics.Data.DataAccess.DataSets.VisitsRow .....	16
2.9	Sitecore.Analytics.Data.DataAccess.DataSets.PagesRow .....	18
2.10	Sitecore.Analytics.Data.DataAccess.DataSets.PageEventsRow .....	20
2.11	Sitecore.Analytics.Data.DataAccess.DataSets.AutomationStatesRow .....	21
2.12	Sitecore.Analytics.Data.DataAccess.DataSets.GeolpsRow .....	22
2.13	Sitecore.Analytics.Data.DataAccess.DataSets.LocationsRow .....	23
2.14	Sitecore.Analytics.Data.DataAccess.DataSets.CampaignsRow .....	24
2.15	Sitecore.Analytics.Data.DataAccess.DataSets.ProfilesRow .....	25

# Chapter 1

## Introduction

This guide describes the Sitecore Engagement Analytics API. It is useful for developers who are looking for information about the Analytics API. It describes the class functionality and important properties/methods.

This document contains the following chapters:

- **Chapter 1 — Introduction**  
This chapter is an introduction to the guide.
- **Chapter 2 — The Analytics API**  
This chapter is an API reference.

## Chapter 2

# The Analytics API

This chapter describes the most frequently used classes in the Sitecore Engagement Analytics API. Each section in this chapter presents an important class in the API. Each section contains the class diagram and the tables that describe the properties and methods of the class.

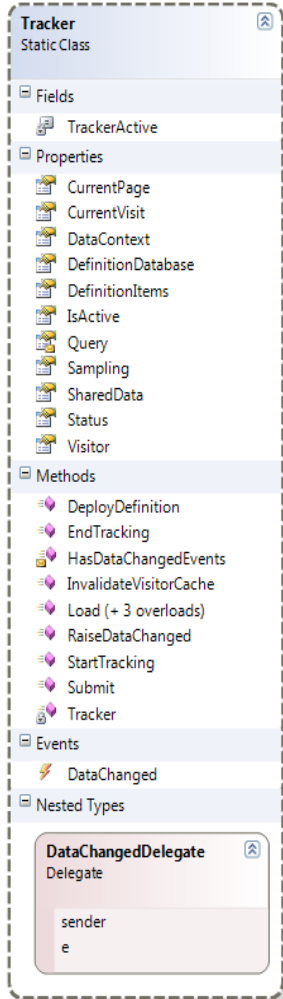
This chapter contains the following sections:

- Sitecore.Analytics.Tracker
- Sitecore.Analytics.Data.DataAccess.Visitor
- Sitecore.Analytics.Visit
- Sitecore.Analytics.TrackerDataContext
- Sitecore.Analytics.Data.TrackerChanges
- Sitecore.Analytics.Data.DataAccess.DataAdapters.SqlBase
- Sitecore.Analytics.Data.DataAccess.DataSets.VisitorsRow
- Sitecore.Analytics.Data.DataAccess.DataSets.VisitsRow
- Sitecore.Analytics.Data.DataAccess.DataSets.PagesRow
- Sitecore.Analytics.Data.DataAccess.DataSets.PageEventsRow
- Sitecore.Analytics.Data.DataAccess.DataSets.AutomationStatesRow
- Sitecore.Analytics.Data.DataAccess.DataSets.GeolpsRow
- Sitecore.Analytics.Data.DataAccess.DataSets.LocationsRow
- Sitecore.Analytics.Data.DataAccess.DataSets.CampaignsRow
- Sitecore.Analytics.Data.DataAccess.DataSets.ProfilesRow

## 2.1 Sitecore.Analytics.Tracker

The `Sitecore.Analytics.Tracker` is a static class that is used to track each visitor. You do not have to construct it or retrieve it from a factory or a provider. It provides access to the Analytics context.

The most important properties of the `Tracker` class are:



Property	Description
<code>static Visitor</code> Visitor	Represents the visitor that is tracked during a visit.
<code>static VisitorDataSet.Visit</code> <code>sRow CurrentVisit</code>	Is an instance of the <code>Sitecore.Analytics.Data.DataAccess.Visit</code> or class.  Represents the current visit. It corresponds directly to a record in the database represented by the <code>VisitorDataSet.VisitsRow</code> class.  Is an instance of the <code>VisitorDataSet.VisitsRow</code> class.
<code>static VisitorDataSet.Pages</code> <code>Row CurrentPage</code>	Represents the current page request.
<code>static TrackerDataContext</code> DataContext	Exposes utility methods for working with visit data.
<code>static bool</code> IsActive	Indicates whether the Analytics add-on is enabled for the current request.
<code>static Database</code> DefinitionDatabase	Gets the definition database.
<code>static AnalyticsItems</code> DefinitionItems	Gets the definition items.  Returns a reference to the analytics definitions in the <code>DefinitionDatabase</code> property.

The most important methods of the `Tracker` class are:

Method	Description
<code>static void Submit()</code>	<p>Submits the changes made in the current context to the analytics database.</p> <p><b>Note</b> The Engagement Analytics automatically submits every change made during a request at the end of the request. Therefore, it is not necessary to call this method in an HTTP Request context.</p>
<code>static void StartTracking()</code>	<p>The <code>StartTracking</code> pipeline initializes the <code>Tracker</code> class with information about the current visit and page view.</p> <p>It also processes query string parameters that trigger page events or campaigns.</p> <p>This method can also be used to call the <code>ParseReferrer</code> pipeline to associate the visit with a search engine and search keywords. Processors receive an argument of type <code>StartTrackingArgs</code>.</p>
<code>static void EndTracking()</code>	<p>Ends the tracking process, sets the <code>IsActive</code> property to false and submits the changes to the visitor dataset that was created during the page request.</p>
<code>static void Load(VisitorLoadOptions visitorLoadOptions)</code>	<p>These methods load data about previous visits. They load the data in the dataset of the <code>Visitor</code> class.</p> <p><b>Note</b> <code>LoadOptions</code> is a wrapper class around <code>VisitorLoadOptions</code> that provides more convenient API to</p>
<code>static void Load(LoadOptions loadOptions)</code>	
<code>static void Load(SharedDataLoadOptions sharedDataLoadOptions)</code>	

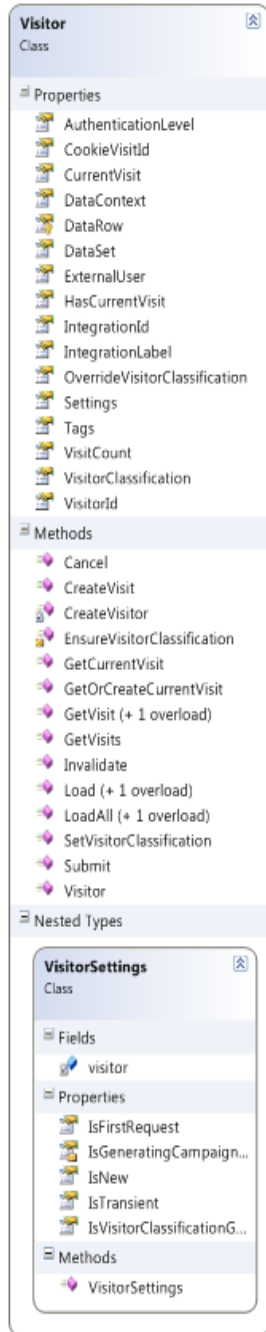
Method	Description
<pre>static void Load(VisitorLoadOptions visitorVisitorLoadOptions, SharedDataLoadOptions sharedDataLoadOptions)</pre>	create the options object.
<pre>static bool DeployDefinition(Item item)</pre>	Deploys the definition from the specified item. Returns <i>true</i> if the deployment was successful.

## 2.2 Sitecore.Analytics.Data.DataAccess.Visitor

The visitor class is used to manage the visits of a certain visitor.

### Note

You can directly access the properties and methods of the `Visitor` class.



The most important properties of the `Visitor` class are:

Property	Description
<code>Guid VisitorId</code>	Is a GUID that is unique for every visitor.
<code>VisitorDataSet.VisitsRow CurrentVisit</code>	Is an instance of the <code>VisitorDataSet.Visits</code> class.  Represents the current visit — corresponds directly to a record in the database represented by the <code>VisitorDataSet.VisitsRow</code> class.
<code>TrackerDataContext DataContext</code>	Exposes utility methods for accessing the visit data.

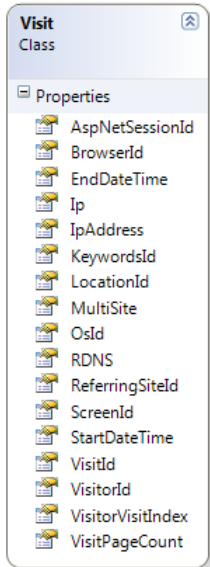
The most important methods of the `Visitor` class are:

Method	Description
<code>void Load(VisitorLoadOptions visitorLoadOptions)</code>	Loads data about previous visits — Loads the data in the dataset of the <code>Visitor</code> class.
<code>void LoadAll(VisitLoadOptions visitLoadOptions, VisitorOptions visitorOptions)</code>	Loads data about all the visits.
<code>virtual void Submit()</code>	Accepts the changes and commits them in the database.
<code>virtual void Cancel()</code>	Rejects the changes.



## 2.3 Sitecore.Analytics.Visit

A visit represents a session. It is the sequence of pages accessed by a single Web client during a single browsing session, including any events associated with those pages. For more information about the visits, see the section *Sitecore.Analytics.Data.DataAccess.DataSets.VisitsRow*.

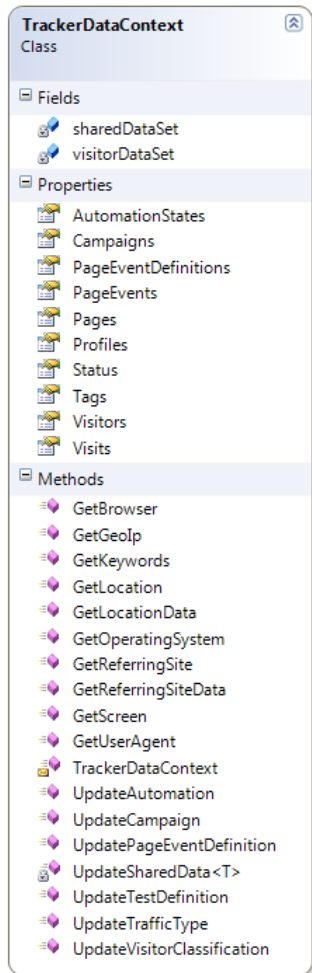


The most important properties of the `Visit` class are:

Property	Description
<code>Guid VisitId</code>	Is a GUID that is unique for each visit.
<code>Guid VisitorId</code>	Is a GUID that is unique for each visitor.
<code>Guid BrowserId</code>	Is a GUID that is unique for each browser such as Internet Explorer, Mozilla and Google Chrome.
<code>Guid ScreenId</code>	Is a GUID that is unique for each screen such as the settings 1024x768 and 1280x900.
<code>Guid ReferringSiteId</code>	Is a GUID that is unique for each site the visitor is redirected from.
<code>Guid OsId</code>	Is a GUID that is unique for each operating system.
<code>Guid LocationId</code>	Is a GUID that is unique for each location the visitor belongs to.
<code>int VisitorVisitIndex</code>	Used to number the visits made by an individual visitor.
<code>int VisitPageCount</code>	Is a counter for each page that counts the number of visits to each page that is registered to be counted.
<code>byte[] Ip</code>	The IP of the visitor.
<code>DateTime StartDateTime</code>	The start time of the visit.
<code>DateTime EndDateTime</code>	The end time of the visit.

## 2.4 Sitecore.Analytics.TrackerDataContext

The `TrackerDataContext` is the visitor context querying class. It is used to get and update the properties of the current visitor.



The most important properties of the `TrackerDataContext` class are:

Property	Description
<code>IEnumerable&lt;VisitorDataSet.VisitorTagsRow&gt; Tags</code>	<p>Stores analytics attributes about visitors. For more information about the visitors, see the section <i>Sitecore.Analytics.Data.DataAccess.DataSets.VisitorsRow</i>.</p> <p><b>Important</b> Analytics tags differ from security profiles and analytics profiles<sup>1</sup>. Analytics tags apply to visitors rather than visits.</p> <p><b>Note</b> Analytics tag values apply to visitors. If a visitor uses two different user agents to access the Web site, the analytics engine records two sessions with separate tag values.</p>
<code>IEnumerable&lt;VisitorDataSet.ProfilesRow&gt; Profiles</code>	<p>See the Profiles description in the section <i>Sitecore.Analytics.Data.DataAccess.DataSets.VisitsRow</i>.</p>

The most important methods of the `TrackerDataContext` class are:

Method	Description
<code>VisitorDataSet.BrowsersRow GetBrowser(string majorName, string minorName, string version)</code>	<p>Gets or creates the specified browser record in the memory dataset.</p>

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

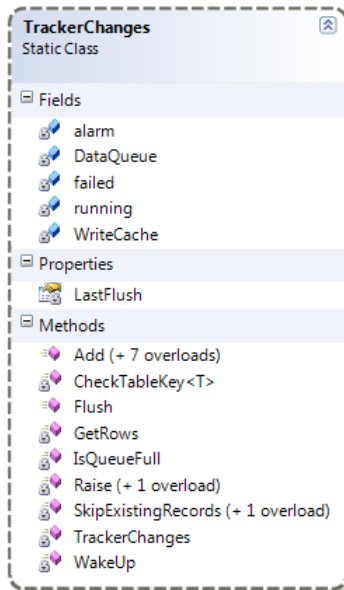
Method	Description
<pre>VisitorDataSet.GeoIpsRow GetGeoIp(byte[] ip)</pre>	Gets or creates the GeoIP record in the memory dataset for the specified IP address.
<pre>VisitorDataSet.KeywordsRow GetKeywords(string keywords)</pre>	Gets or creates the search keyword record in the memory dataset.
<pre>VisitorDataSet.LocationsRow GetLocation(string name, string country)</pre>	Gets or creates the specified location record in the memory dataset.
<pre>VisitorDataSet.LocationsRow GetLocationData(Guid locationId)</pre>	Gets the location record from the memory dataset identified by ID.
<pre>VisitorDataSet.OSRow GetOperatingSystem(string majorName, string minorName, string version)</pre>	Gets or creates the specified Operating System record in the memory dataset.
<pre>VisitorDataSet.ReferringSitesRow GetReferringSite(string host)</pre>	Gets or creates the specified referring site record in the memory dataset.
<pre>VisitorDataSet.ReferringSitesRow GetReferringSiteData(Guid referringSiteId)</pre>	Gets the referring site record by ID.
<pre>VisitorDataSet.ScreensRow GetScreen(string dimensions)</pre>	Gets or creates the screen record in the memory dataset for the specified screen resolution.
<pre>VisitorDataSet.UserAgentsRow GetUserAgent(string userAgent)</pre>	Gets or creates the specified user agent record in the memory dataset. A user agent is any software that is used to access the website such as web browsers, E-mail clients, RSS readers, download managers and robots.

## 2.5 Sitecore.Analytics.Data.TrackerChanges

The `TrackerChanges` is a static class that acts as a queue for all of the changes submitted by the pipeline through the `tracker` class. The main reason for engaging the `TrackerChanges` class in this process is to minimize the database access occurrences. Any visitor related property that is changed must be queued first in the `TrackerChanges` before it is committed to the database later on according to the set rules.

### Note

This class is rarely used by the Analytics developers. It is used to improve the performance.



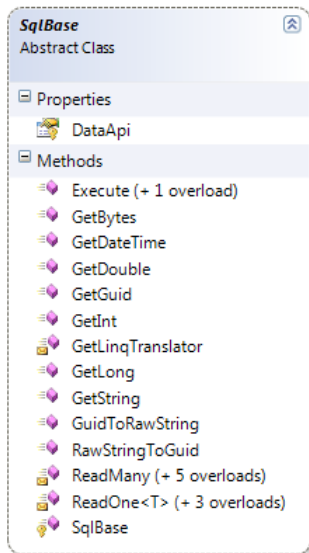
## 2.6 Sitecore.Analytics.Data.DataAccess.DataAdapters.SqlBase

The `SqlBase` is an abstract class that contains methods which developers use to build custom queries on the Analytics database. You can access the instance of this class implementation using the following static property:

```
Sitecore.Analytics.Data.DataAccess.DataAdapters.DataAdapterManager.Sql
```

The standard data provider implements this class via

```
Sitecore.Analytics.Data.DataAccess.DataAdapters.Sql.Sql
```



The most important methods of the `SqlBase` class are:

Method	Description
<pre>virtual int Execute(string commandText, params object[] parameters)</pre>	Executes the query and returns the number of rows affected.
<pre>virtual T ReadOne&lt;T&gt;(string commandText, Func&lt;DataProviderReader, T&gt; selector, Func&lt;T&gt; defaulter, params object[] parameters)</pre>	Reads one instance of type T.
<pre>virtual List&lt;T&gt; ReadMany&lt;T&gt;(string commandText, Func&lt;DataProviderReader, T&gt; initializer, int maxEntries, params object[] parameters)</pre>	Reads many instances of type T.

### Note

To make your queries independent of the SQL provider, you can use the *pseudo SQL* syntax provided by the methods in the `SqlBase` class.

### Pseudo SQL Syntax

The main difference between normal SQL expressions and pseudo SQL expressions is that:

In pseudo SQL expressions, you must wrap all table and field names in special symbols which Sitecore then uses to insert appropriate quotes and brackets into the SQL expression. For instance:

- You must use the symbols `{0}` and `{1}` around all table and field names.

#### Example

```
"select {0}Campaigns{1}.{0}CampaignId{1} from {0}Campaigns{1}"
```

- You must use the symbols `{2}` and `{3}` around parameters.

#### Example

```
"where {0}CampaignId{1} = {2}campaignId{3}"
```

- You must use the symbol {4} to insert a quote and symbol {5} to insert a wildcard.

**Example**

```
"where {0}Path{1} LIKE {4}/sitecore/{5}{4}"
```

- You must use the symbol {6} to insert an empty string literal.

**Example**

```
"where {0}Value{1} != {6}"
```

**Note**

To get the Visit ID of the latest registered visit in DMS, use the `ReadOne` method:

```
string query = "select
top 1 {0}Visits{1}.{0}VisitId{1} from {0}Visits{1} order by {0}StartDateTime{1} desc;"
Guid visitorId = DataAdapterManager.Sql.ReadOne<Guid>(query, reader =>
DataAdapterManager.Sql.GetGuid(0, reader), new object[0]);
```

To get the list of the IDs of the latest visits where any failures were registered, use the `ReadMany` method:

```
string query = "select distinct {0}Pages{1}.{0}VisitId{1}"
+ "from {0}Pages{1}, {0}PageEvents{1}, {0}PageEventDefinitions{1}"
+ "where "
+ "DATEDIFF(day, {0}Pages{1}.{0}DateTime{1}, getdate()) < 1 and"
+ "{0}PageEvents{1}.{0}PageId{1} = {0}Pages{1}.{0}PageId{1} and"
+ "{0}PageEventDefinitions{1}.{0}PageEventDefinitionId{1} = "
+ "{0}PageEvents{1}.{0}PageEventDefinitionId{1} and"
+ "{0}PageEventDefinitions{1}.{0}IsFailure{1} = 1";
List<Guid> visitsId = DataAdapterManager.Sql.ReadMany<Guid>(query, reader =>
DataAdapterManager.Sql.GetGuid(0, reader), new object[0]);
var listResult = new StringBuilder();
visitsId.ForEach(g => listResult.Append(g).Append(", "));
Page.Response.Output.WriteLine("visitsID: [" + listResult.ToString() + "]);
```

To get the list of visited pages for a specific visit, use the `ReadMany` method with the corresponding Visit ID passed as a parameter:

```
string query = "select {0}Pages{1}.{0}Url{1}"
+ "from {0}Pages{1}"
+ "where "
+ "{0}Pages{1}.{0}VisitId{1} = {2}visitId{3}";
List<string> visitedPages = DataAdapterManager.Sql.ReadMany<string>(query, reader =>
DataAdapterManager.Sql.GetString(0, reader), new object[] { "visitId",
Tracker.CurrentVisit.VisitId });
var listResult = new StringBuilder();
visitedPages.ForEach(g => listResult.Append(g).Append(", "));
Page.Response.Output.WriteLine("URLs: [" + listResult.ToString() + "]);
```

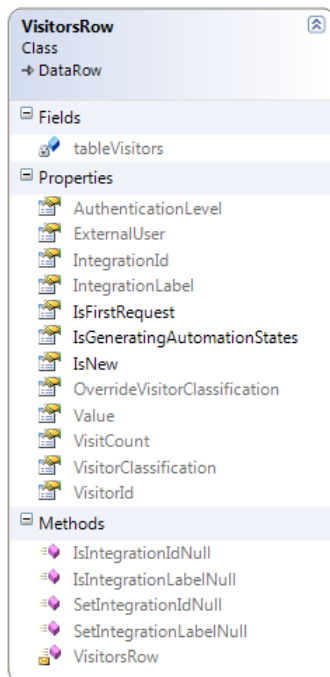
## 2.7 Sitecore.Analytics.Data.DataAccess.DataSets.VisitorsRow

The `VisitorsRow` is a partial class within the `VisitorsDataSet` class. You can use this class to access the data gathered about any visitor in the tracker. For more information about the visitors, see the section *Sitecore.Analytics.Data.DataAccess.Visitor*.

A visitor represents multiple individual visits from a single user agent. For more information about the visits, see the section *Sitecore.Analytics.Data.DataAccess.DataSets.VisitsRow*.

### Note

The web client cookie allows visitors sessions to expire in one year.

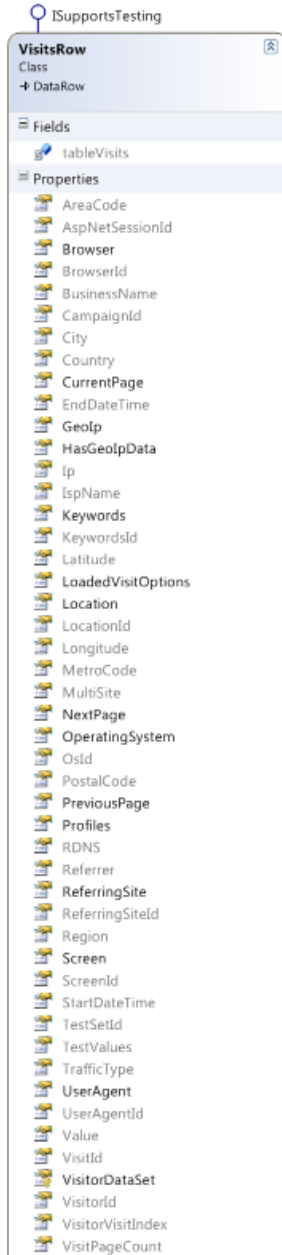


The most important properties of the `VisitorsRow` class are:

Property	Description
<code>bool IsFirstRequest</code>	Sets or gets a value that indicates whether this instance is the first request in the row.
<code>bool IsNew</code>	Sets or gets a value that indicates whether this instance is a new request in the row.

## 2.8 Sitecore.Analytics.Data.DataAccess.DataSets.VisitsRow

The `VisitsRow` is a partial class within the `VisitorsDataSet` class. You can use this class to access the data gathered for any visit in the tracker through this class. It implements the `ISupportsTesting` interface. For more information about the visits, see the section *Sitecore.Analytics.Visit*.

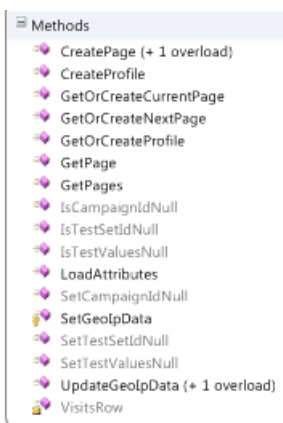


The most important properties of the `VisitsRow` class are:

Property	Description
<code>IEnumerable&lt;ProfilesRow&gt; Profiles</code>	<p>Contains information gathered during each analytics' visit. You can assign analytics profile values to content items and automatically update the user's analytics profile when the user accesses that content. For more information about the visits, see the section <i>Sitecore.Analytics.Visit</i>.</p> <p><b>Important</b> Analytics profiles differ from security profiles and Analytics tags<sup>2</sup>. Analytics profiles apply to individual visits in individual web clients rather than to users or visitors. For more information about analytics tags, see the description of the <code>Tags</code> property in the section <i>Sitecore.Analytics.TrackerDataContext</i>.</p> <p><b>Note</b> The Analytics profile values apply to individual visits in individual web clients. If a visitor uses two different user agents to accesses the web site, the analytics engine records two visits with separate profile values.</p>

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





The most important methods of the `VisitsRow` class are:

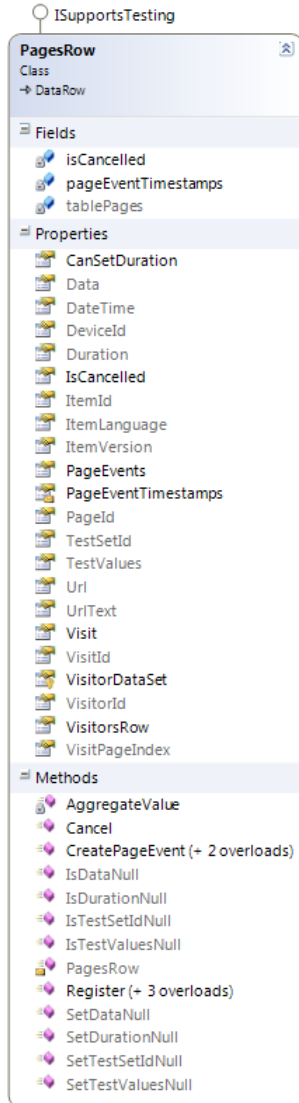
Method	Description
<pre>virtual PagesRow CreatePage ()</pre>	Unconditionally creates a new page row in the memory dataset and makes it the current page of the request.
<pre>ProfilesRow CreateProfile (string profileName)</pre>	Creates a new profile row for the specified profile name.
<pre>PagesRow GetOrCreateCurrentPage ()</pre>	Gets the page row for the current page and creates the row if it does not exist.
<pre>PagesRow GetOrCreateNextPage ()</pre>	Gets the page row for the "next" page and creates the row if it does not exist.
<pre>ProfilesRow GetOrCreateProfile (string profileName)</pre>	Gets the profile row for the specified profile and creates the row if it does not exist.
<pre>virtual PagesRow GetPage (int pageNumber)</pre>	Gets the page with the specified page number.
<pre>virtual IEnumerable&lt;PagesRow&gt; GetPages ()</pre>	Gets all the pages for this visit.
<pre>void LoadAttributes ()</pre>	Loads the attributes of the visit that are stored in separate tables such as the user agents, browser, operating system, screen, keywords and referring site.
<pre>bool UpdateGeoIpData ()</pre>	Updates the visit GeoIP data if it is available. Returns <i>true</i> if data was updated successfully or if the data is already up-to-date.

## 2.9 Sitecore.Analytics.Data.DataAccess.DataSets.PagesRow

The `PagesRow` is a partial class within the `VisitorsDataSet` class and implements the `ISupportsTesting` interface. You can access the data gathered about any visitor in the tracker through this class.

A page represents a page in a session. You can associate events with the previous, current, and next pages in the user's visit.

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](#).



The most important properties of the `PagesRow` class are:

Property	Description
<code>bool CanSetDuration</code>	Gets or sets a value indicating whether this instance can set the time that the visitor spent viewing a page. The duration can be set for all regular pages. The Media and the <code>VisitorIdentification.aspx</code> page should not set duration. <i>True:</i> If this instance can set the duration.
<code>bool IsCancelled</code>	Gets a value indicating whether this page is cancelled. <i>True:</i> If this page instance is cancelled.
<code>IEnumerable&lt;PageEventsRow&gt; PageEvents</code>	Gets the page events.
<code>VisitsRow Visit</code>	Gets the visit.
<code>VisitorsRow VisitorsRow</code>	Gets the visitors row.

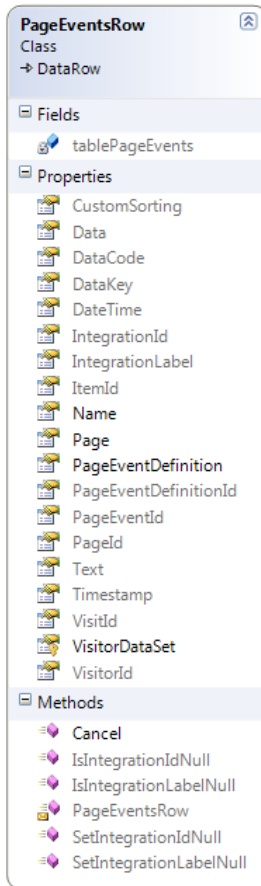
The most important methods of the `PageRow` class are:

Method	Description
<pre>virtual PageEventsRow CreatePageEvent (PageEventDefinitionsData pageEventDefinition)</pre>	<p>Creates a page event and adds it to a page event row instance.</p>
<pre>PageEventsRow Register (PageEventData data)</pre>	<p>Associates a page event with a page and updates the corresponding page event row instance.</p> <p>Parameter: <code>data</code> — all the attributes that can be stored with a page event.</p> <p><b>Example</b> You can use the following snippet to register the event <code>Rate</code> and fill its attributes using the <code>PageEventData</code> class.</p> <pre>PageEventData ped = new     PageEventData ("Rate") {     Text = "simple text",     Data = "simple data",     DataKey = "simple datakey", }; Tracker.CurrentPage.Register (ped);</pre>
<pre>PageEventsRow Register (PageEventItem eventItem)</pre>	<p>Associates a page event with a page and updates the corresponding page event row instance.</p> <p>Parameter: <code>eventItem</code> — the event item to be associated.</p>
<pre>PageEventsRow Register (string name, string text)</pre>	<p>Associates a page event with a page and updates the corresponding page event row instance.</p> <p>Parameters:</p> <ul style="list-style-type: none"> <li><code>name</code> — the name of the page event</li> <li><code>text</code> — the associated textual description.</li> </ul>
<pre>void Cancel ()</pre>	<p>Cancels a page from being tracked and sets the <code>IsCancelled</code> property to <code>true</code>.</p>

## 2.10 Sitecore.Analytics.Data.DataAccess.DataSets.PageEventsRow

The `PageEventsRow` is a partial class within the `VisitorsDataSet` class. You can access the data gathered for any visitor in the tracker through this class.

A page event represents an event, goal, or failure associated within an Analytics page.



The most important property of the `PageEventsRow` class is:

Property	Description
<code>string</code> Name	Gets the name of the page event definition.

The most important method of the `PageEventsRow` class is:

Method	Description
<code>void</code> Cancel ()	<p>Cancels a page event from being tracked.</p> <p>Rejects all the changes made to the page events row since <code>AcceptChanges</code>. <code>AcceptChanges</code> is a nested method within the <code>CreatePageEvent</code> method of the <code>PagesRow</code> class.</p>

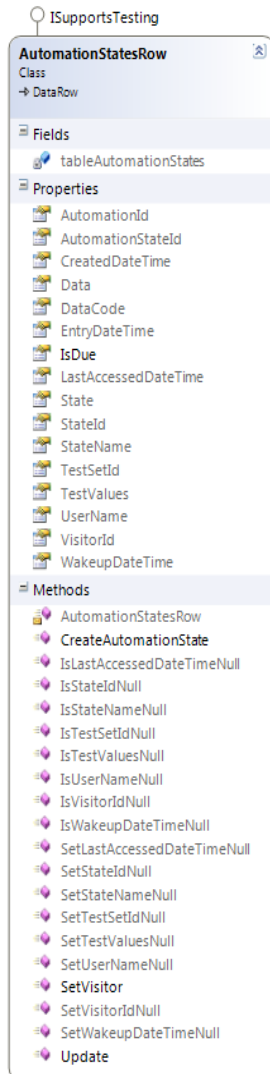
## 2.11 Sitecore.Analytics.Data.DataAccess.DataSets.AutomationStatesRow

The `AutomationStatesRow` is a partial class within the `VisitorsDataSet` class. You can access the data gathered about any visitor in the tracker through this class. It implements the `ISupportsTesting` interface.

**Note**

You will rarely use the `AutomationStates` class directly.

The most important properties of the `AutomationStatesRow` class are:



Property	Description
<code>System.Guid</code> <code>AutomationStateId</code>	Is a GUID that is unique for every automation state.
<code>bool</code> <code>IsDue</code>	Set or gets a value indicating whether this automation state is due.  Returns <i>true</i> if the instance is due.

The most important methods of the `AutomationStatesRow` class are:

Method	Description
<code>static</code> <code>AutomationStatesRow</code> <code>CreateAutomationState</code> <code>(AutomationStatesData</code> <code>Table,</code> <code>Guid</code> <code>automationStateId)</code>	Creates an automation state.
<code>void</code> <code>SetVisitor</code> <code>(Visitor</code> <code>visitor)</code>	Binds a specific visitor to the automation state.

```

static
AutomationStatesRow
Update (AutomationSta
tesDataTable
automationStatesData
Table, Guid
automationStateId,
Guid automationId,
DateTime dateTime,
string data, Guid
stateId)

```

Updates a specified automation state.

The properties to be updated are:

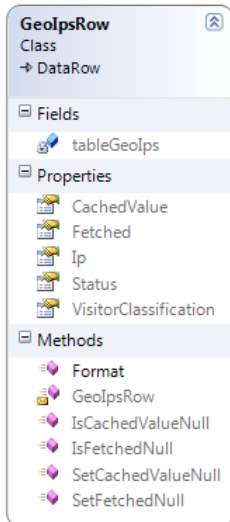
- AutomationStateId
- AutomationId
- CreatedDateTime
- Data
- StateId

## 2.12 Sitecore.Analytics.Data.DataAccess.DataSets.GeoIpsRow

The `GeoIpsRow` is a partial class within the `VisitorsDataSet` class. This class acts as a caching table.

### Note

You will rarely use the `GeoIpsRow` class directly. The GeoIP information is available on the `VisitsRow` class.

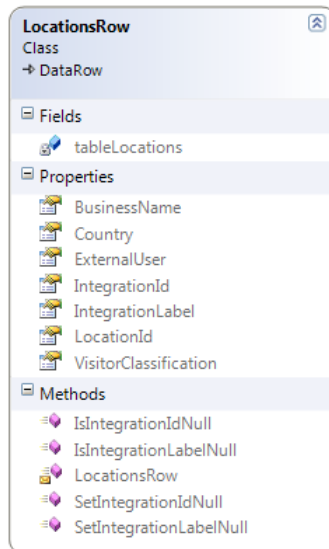


## 2.13 Sitecore.Analytics.Data.DataAccess.DataSets.LocationsRow

The `LocationsRow` is a partial class within the `VisitorDataSet` class. This class acts as a reporting table.

### Note

You will rarely use the `LocationsRow` class directly. The `LocationsRow` information is available on the `VisitsRow`.

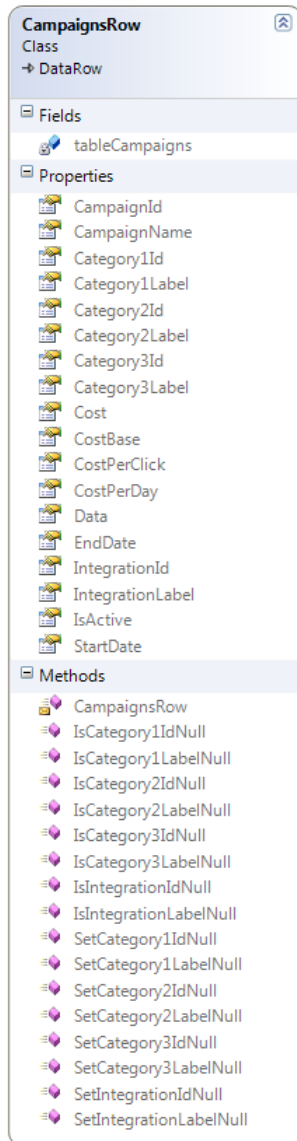


## 2.14 Sitecore.Analytics.Data.DataAccess.DataSets.CampaignsRow

The `CampaignsRow` is a partial class within the `SharedDataSet` class. This class acts as a reporting table that contains campaign information.

A campaign is a marketing effort to promote awareness of a product, service, or other offering through one or more media channels. The Analytics Engagement users can filter analytics data by campaign.

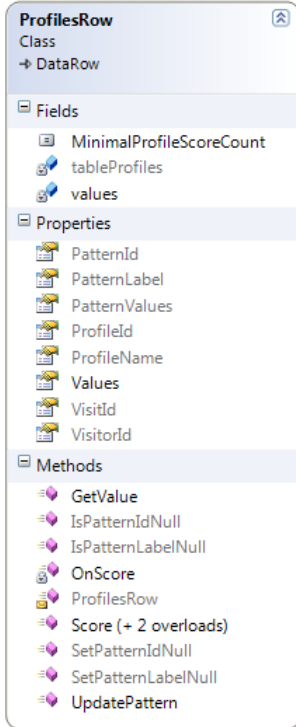
The developer can access the campaign information by using the `CampaignId` property of the `VisitsRow` class.





## 2.15 Sitecore.Analytics.Data.DataAccess.DataSets.ProfilesRow

The ProfilesRow is a partial class within the VisitorsDataSet class.



The most important property of the ProfilesRow class are:

Property	Description
VisitProfile Values	Is a storage for key/value pairs. For example: OrganicFood=10, Meat=200, Vegetables=10 and Dairy=50.

The most important methods of the ProfilesRow class are:

Method	Description
Float GetValue (string profileKeyName)	Gets the Key/value pair of a profile key.
void Score (Dictionary<string, float> profile)	These methods score a specific profile.
void Score (string key, float value)	
void Score (ContentProfile profileName)	