By default, web analytics log data for all visitors on the live website. Developers can customize the system to log web analytics only for visitors who give some form of consent.

The system provides the **WebAnalyticsEvents.CheckAnalyticsConsent** global event, which triggers before any type of web analytics are logged for visitors (before processing of visitor personal data). By assigning a handler to the event, developers can perform the following:

- Define a condition according to the website's consent requirements. The condition can be based on consents managed within Kentico or on any other data available in the context of the request.
- Set the **HasConsent** property of the handler's **CheckAnalyticsConsentEventArgs** parameter based on the result. If the property is *false*, the system does not process web analytics for the given request.

Example

The following example demonstrates how to disable logging of web analytics for all visitors (contacts) who have not accepted a consent with the *WebAnalytics* code name.

> **Kentico EMS required**
>
> The example uses the contact tracking and consent features to evaluate whether a visitor has given consent to log web analytics. As a result, it only works for sites with a **Kentico EMS** license.

1. Open your Kentico project in Visual Studio (using the **WebSite.sln** or **WebApp.sln** file).
2. Create a custom module class.
   - We recommend adding the class into a custom *Class library* project within the Kentico solution.
3. Override the module's **OnInit** method and assign a handler method to the **WebAnalyticsEvents.CheckAnalyticsConsent.Execute** event.
4. Perform the required actions within the handler method:
   - To disable logging of web analytics for a given request, set the **HasConsent** property of the handler's **CheckAnalyticsConsentEventArgs** parameter to *false*.
   - You can evaluate whether to log web analytics based on any custom condition. Access data from request contexts or use the **ContactId** property of the handler's **CheckAnalyticsConsentEventArgs** parameter to get the ID of the contact object for which the analytics are being logged.

```
using CMS;
using CMS.DataEngine;
using CMS.Core;
using CMS.DataProtection;
using CMS.ContactManagement;
using CMS.WebAnalytics;

// Registers the custom module into the system
[assembly: RegisterModule(typeof(CustomConsentModule))]

public class CustomConsentModule : Module
{
    // Module class constructor, the system registers the module under the name
"CustomConsent"
    public CustomConsentModule()
        : base("CustomConsent")
    {
    }

    // Contains initialization code that is executed when the application starts
    protected override void OnInit()
    {
        base.OnInit();

        // Assigns a handler to the WebAnalyticsEvents.CheckAnalyticsConsent.Execute
event
        // This event occurs before the system logs web analytics for visitors and
processes their personal data
        WebAnalyticsEvents.CheckAnalyticsConsent.Execute += Analytics_CheckConsent;
    }

    private void Analytics_CheckConsent(object sender, CheckAnalyticsConsentEventArgs
e)
    {
        // Gets the contact for which the analytics are being logged
        // Note: The contact is null for visitors who have not given consent for
contact tracking
        ContactInfo contact = ContactInfoProvider.GetContactInfo(e.ContactId);

        // Gets the consent that will be evaluated
        ConsentInfo consent = ConsentInfoProvider.GetConsentInfo("WebAnalytics");

        // Disables web analytics logging for all visitors (contacts) who have not
accepted the "WebAnalytics" consent declaration
        IConsentAgreementService consentService = Service.
Resolve<IConsentAgreementService>();
        e.HasConsent = (contact != null && consent != null) && consentService.IsAgreed
(contact, consent);
    }
}
```

Performance recommendations

The system triggers the **CheckAnalyticsConsent** event frequently when displaying pages to visitors (multiple times per web request in certain cases). Running demanding operations within the event's handler method may have a negative impact on your site's performance.

We strongly recommend caching the results of your consent conditions. To learn about the custom caching options provided by the Kentico API, see: Caching in custom code

Tip – Logging analytics based on tracking consent

For visitors who have not given consent for the tracking of contacts and activities (see Working with consents), the **ContactId** property of the handler's **CheckAnalyticsConsentEventArgs** parameter is 0.

You can use the tracking consent as your condition for logging of web analytics. In this case, you do not need to define or evaluate an additional "web analytics consent". You only need to create a condition that checks whether the *ContactId* value is greater than 0, i.e. that the web analytics are being logged for an existing contact.