


Kentico does not provide any built-in components that give visitors an option to revoke their [consent agreements](#).

To comply with the [requirements of the GDPR](#), you need to develop and maintain custom components that allow visitors to revoke consent agreements on your website. This way, you can create a solution that best reflects the state of [personal data](#) processing on your website.

 "It shall be as easy to withdraw as to give consent."

(Source: [GDPR Article 7](#), Paragraph 3)

To manage consent agreements in the code of your custom components, use the default implementation of the **IConsentAgreementService** interface. The interface provides the following methods:

- **GetAgreedConsents(ContactInfo)** – returns a collection of all consents with which the specified contact has agreed. The returned *Consent* objects provide the *GetConsentText* method, which automatically gets either the current texts of the given consent or the texts of the archived consent version for which the agreement was given.
- **Agree(ContactInfo, ConsentInfo)** – creates a consent agreement for a specified contact and consent.
- **Revoke(ContactInfo, ConsentInfo)** – revokes the consent agreement for a specified contact and consent (if such an agreement exists).
- **IsAgreed(ContactInfo, ConsentInfo)** – evaluates whether a specified contact has agreed with a given consent.

Clearing personal data for revoked consents

When a visitor revokes a consent agreement, you may also need to delete or anonymize certain types of [personal data](#) stored by the system.

To perform additional actions of this type whenever a consent agreement is revoked, assign a custom handler to the system's **DataProtectionEvents.RevokeConsentAgreement** [global event](#).

See also: [Implementing personal data erasure](#)

Revoking tracking consent through the API

Consent with the [tracking of contacts](#) and their [activities](#) is controlled by the [cookie level](#) allowed by each visitor.

By default, visitors can manage their tracking consent and cookie level through the **Cookie law and tracking consent** web part. However, if you wish to revoke the tracking consent through the API, you also need to implement code that lowers the visitor's cookie level (below the *Visitor* level).

```
using CMS.Core;
using CMS.Helpers;

...

ICurrentCookieLevelProvider cookieLevelProvider = Service.
Resolve<ICurrentCookieLevelProvider>();
cookieLevelProvider.SetCurrentCookieLevel(cookieLevelProvider.
GetDefaultCookieLevel());
```

Example – Creating a web part for revoking consent agreements

The following example demonstrates how to [create a web part](#) that displays all consents with which the current visitor (contact) has agreed and allows them to be revoked.

Start by implementing the web part code:



1. Open your web project in Visual Studio using the **WebSite.sln** (or **WebApp.sln**) file.
2. Right-click the **CMSWebParts** folder in the Solution Explorer and click **Add -> New Folder**.
3. Name the sub-folder *CustomWebParts*.
4. Create a **Web User Control** named *RevokeConsents.ascx* in the *CustomWebParts* folder.
5. Copy the following code into the control's markup:

```
<%@ Control Language="C#" AutoEventWireup="true"
    CodeFile="~/CMSWebParts/CustomWebParts/RevokeConsents.
ascx.cs"
    Inherits="CMSWebParts_CustomWebParts_RevokeConsents" %>

<cms:LocalizedLabel ID="lblInfo" runat="server"
    Text="Consent was successfully revoked."
    EnableViewState="false"
    Visible="false" />
<cms:LocalizedLabel ID="lblNoData" runat="server"
    Text="You have not given any consents yet."
    EnableViewState="false" />

<asp:Repeater runat="server" ID="rptConsents">
    <ItemTemplate>
        <h3 class="consent-heading">
            <%# HttpUtility.HtmlEncode(Eval("ConsentDisplayName")) %>
        </h3>
        <cms:LocalizedButton runat="server" ID="btnRevoke"
            Text="Revoke"
            CommandArgument='<%# Eval("ConsentID") %>'
            OnClick="btnRevoke_Click" />

        <p>
            <%# Eval("ConsentShortText") %>
        </p>
    </ItemTemplate>
</asp:Repeater>
```

6. Open the control's code behind file (*RevokeConsents.ascx.cs*) and use the following code:

```
using System;
using System.Linq;
using System.Collections.Generic;
using System.Web.UI.WebControls;

using CMS.ContactManagement;
using CMS.Core;
using CMS.DataProtection;
using CMS.Helpers;
using CMS.Localization;
using CMS.PortalEngine.Web.UI;

public partial class CMSWebParts_CustomWebParts_RevokeConsents :
    CMSAbstractWebPart
{
    private IConsentAgreementService consentAgreementService;
    private ContactInfo currentContact;

    // Actions performed when the page is loaded
    public override void OnContentLoaded()
    {
        base.OnContentLoaded();
    }
}
```



```
// Prepares an instance of the default IConsentAgreementService
implementation for managing consents
consentAgreementService = Service.Resolve<IConsentAgreementService>();

// Gets the current contact
currentContact = ContactManagementContext.CurrentContact;

LoadConsentData();
}

// Loads and displays the consent agreements for the current contact
private void LoadConsentData()
{
    // Does not attempt to load consent data if the current contact is not
    available
    // This occurs if contact tracking is disabled or for visitors who have
    not agreed with the tracking consent
    if (currentContact == null)
    {
        return;
    }

    // Gets all consents with which the current contact has agreed
    IEnumerable<Consent> consents = consentAgreementService.GetAgreedConsents
    (currentContact);

    // Prepares a data source for the internal repeater control from the
    loaded consents
    var dataSource = consents.Select(consent => new
    {
        // Each data item contains the ID, display name and short text of the
        given consent
        ConsentID = consent.Id,
        ConsentDisplayName = consent.DisplayName,
        ConsentShortText = consent.GetConsentText(LocalizationContext.
        CurrentCulture.CultureCode).ShortText
    });

    // Binds the data source to the repeater
    rptConsents.DataSource = dataSource;
    rptConsents.DataBind();

    // Displays a message if the current contact has not agreed with any
    consents
    lblNoData.Visible = !consents.Any();
}

// Revokes a specified consent agreement for the current contact
protected void btnRevoke_Click(object sender, EventArgs e)
{
    // Gets the consent based on an ID parameter provided by the clicked
    revoke button
    Button btn = (Button)sender;
    int consentId = ValidationHelper.GetInteger(btn.CommandArgument, 0);
    ConsentInfo consent = ConsentInfoProvider.GetConsentInfo(consentId);

    if (consent != null)
    {
        // Revokes the specified consent and displays a message to inform the
```

```
user
    consentAgreementService.Revoke(currentContact, consent);
    LoadConsentData();
    lblInfo.Visible = true;
}
}
```

7. Save the new files (if you have a web application installation, build the *CMSApp* project).

After creating the web part code, you need to [register the web part in the system](#) (you do not need to set up any web part properties) and [add an instance of the web part](#) to a page on your website.

To test the functionality of the web part:

1. Open your live site in a different browser or an incognito/private window.
2. Agree with some [consents](#).
3. Navigate to the page with the web part.

The web part displays a list of the accepted consents, and buttons that allow you to revoke individual consents.

Rental form

Revoke

I hereby accept that these provided information can be used for marketing purposes and targeted website content.

Tracking consent

Revoke

We have exciting offers and news about our products and services that we hope you'd like to hear about. To present you the offers that suit you the most, we need to know a few personal details about you. We will gather some of your activities on our website (such as which pages you've visited, etc.) and use them to personalize the website content and improve analytics about our visitors. In addition, we will store small pieces of data in your browser cookies. We promise we will treat your data with respect, store it in a secured storage, and won't release it to any third parties.