

To create a custom notification gateway class:

1. Create a new library (assembly) as part of your web project solution.
2. Create a new class inside this library.
3. Set your class to inherit from the **CMS.Notifications.Web.UI.CMSNotificationGateway** abstract class.
4. Override the following methods to achieve the required functionality:
  - **void SendNotification()** – sends a single notification. Automatically called after the specified event is raised.
  - **CMSNotificationGatewayForm GetNotificationGatewayForm()** – loads and returns the notification gateway form for the notification gateway.
5. Compile the library.
6. Add a reference to the library file (\*.dll) to the main Kentico web project.

## Example - E-mail notification gateway class

The following code sample shows how to implement a custom e-mail notification gateway class:

### EmailNotificationGateway.cs

```
using System;
using System.Text;
using System.Data;
using System.Collections;
using System.Collections.Generic;
using System.Web;
using System.Web.UI;

using CMS.Notifications.Web.UI;
using CMS.Helpers;
using CMS.EventLog;
using CMS.SiteProvider;
using CMS.EmailEngine;
using CMS.DataEngine;

namespace EmailNotification
{
    /// <summary>
    /// Base class for e-mail notification gateway.
    /// </summary>
    public class EmailNotificationGateway : CMSNotificationGateway
    {
        /// <summary>
        /// Returns the e-mail gateway form.
        /// </summary>
        public override CMSNotificationGatewayForm GetNotificationGatewayForm()
        {
            try
            {
                Control ctrl = this.NotificationSubscriptionControl.Page.LoadControl("~/CorporateSite/EmailNotificationForm.ascx");
                ctrl.ID = ValidationHelper.GetIdentifier("notif" + this.NotificationGatewayObj.GatewayName);

                return (CMSNotificationGatewayForm)ctrl;
            }
            catch (Exception ex)
            {
                try
                {
                    // Logs exception events
                }
            }
        }
    }
}
```



```
        EventLogProvider.LogException("EmailGateway", "EXCEPTION", ex);
    }
    catch
    {
        // Unable to log the event
    }
}
return null;
}

/// <summary>
/// Sends the notification via e-mail.
/// </summary>
public override void SendNotification()
{
    try
    {
        if (this.NotificationSubscriptionObj != null)
        {
            // Get template text
            NotificationTemplateTextInfo templateText =
            NotificationTemplateTextInfoProvider.GetNotificationTemplateTextInfo(this.
            NotificationGatewayObj.GatewayID, this.NotificationSubscriptionObj.
            SubscriptionTemplateID);
            if (templateText != null)
            {
                // Get the site name
                string siteName = null;
                SiteInfo si = SiteInfoProvider.GetSiteInfo(this.
                NotificationSubscriptionObj.SubscriptionSiteID);
                if (si != null)
                {
                    siteName = si.SiteName;
                }

                // Create message object
                EmailMessage message = new EmailMessage();

                // Get sender from settings
                message.From = SettingsKeyInfoProvider.GetStringValue(siteName
+ ".CMSSendEmailNotificationsFrom");

                // Do not send the e-mail if there is no sender specified
                if (message.From != "")
                {
                    // Initialize message
                    message.Recipients = this.NotificationSubscriptionObj.
                    SubscriptionTarget;

                    // Body
                    if ((this.NotificationSubscriptionObj.SubscriptionUseHTML)
&&
                        (this.NotificationGatewayObj.GatewaySupportsHTMLText)
&&
                        (templateText.TemplateHTMLText != ""))
                    {
                        // HTML format, set Body property, resolve macros if
possible
                        message.EmailFormat = EmailFormatEnum.Html;
                        if (this.Resolver != null)
```



```
        {
            message.Body = this.Resolver.ResolveMacros
(templateText.TemplateHTMLText);
        }
        else
        {
            message.Body = templateText.TemplateHTMLText;
        }
    }
    else
    {
        // Plaintext format, set PlainTextBody property,
resolve macros if possible
        message.EmailFormat = EmailFormatEnum.PlainText;
        if (this.Resolver != null)
        {
            message.PlainTextBody = this.Resolver.ResolveMacros
(templateText.TemplatePlainText);
        }
        else
        {
            message.PlainTextBody = templateText.
TemplatePlainText;
        }
    }

    // Subject, resolve macros if possible
    if (this.Resolver != null)
    {
        message.Subject = this.Resolver.ResolveMacros
(templateText.TemplateSubject);
    }
    else
    {
        message.Subject = templateText.TemplateSubject;
    }

    // Send email via Email engine API
    EmailSender.SendEmail(siteName, message);
}
}
}
}
}
}
}
}
}
}
}
```

