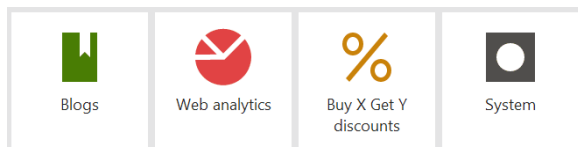


You can modify Kentico [application tiles](#) to display various data. Tiles that display application data are called *Live tiles*. Live tiles display a *numerical value* and a *description* of the numerical value.



Content management

- **Blogs** - displays the number of comments on the site waiting for approval.
- **My blogs** - displays the number of comments waiting for the user's approval.
- **Forms** - displays the number of forms submitted by users in the last 7 days.
- **Pending pages** - displays the number of pages pending the user's approval.
- **Translations** - displays the number of translation submissions waiting for import.

On-line marketing

- **A/B tests** - displays the number of running A/B tests.
- **Contact management** - displays the number of new contacts in the last 7 days.
- **My pending contacts** - displays the number of contacts waiting for the user's action in a Marketing automation process.
- **Web analytics** - displays the number of site visits in the last 7 days.

E-commerce

- **Buy X Get Y discounts** - displays the number of running Buy X Get Y discounts.
- **Catalog discounts** - displays the number of running catalog discounts.
- **Free shipping offers** - displays the number of running free shipping offers.
- **Gift cards** - displays the number of running gift card offers.
- **Order discounts** - displays the number of running order discounts.
- **Product coupons** - displays the number of running product coupon offers.

Configuration

- **Email queue** - displays the number of e-mails that could not be sent.
- **Licenses** - displays days left until license expiration.
- **System** - displays the time since the last application restart.
- **Users** - displays the number of registered users waiting for approval.

You can modify the existing Live tiles to display different data or make static tiles into Live tiles.

Ideally, Live tiles should display information that may require users' attention. By having certain information displayed directly on the application dashboard, users don't have to visit the application itself unless necessary.

1. Create a class that implements the *ILiveTileModelProvider* interface.
2. Implement the logic of the class. The *GetModel* method needs to return the data of the Live tile in one of the following formats:
 - a. *LiveTileModel* object containing:
 - i. **Value** - the numerical value displayed on the Live tile
 - ii. **Description** - the description of the numerical value
 - b. *Null* - in cases when you want the application tile to stay static. For example, when the number of objects displayed by the tile is 0.

```
using System;

using CMS.Activities;
using CMS.ApplicationDashboard.Web.UI;
using CMS.ContactManagement.Web.UI;
using CMS.Core;
using CMS.Helpers;

public class ContactManagementLiveTileModelProvider : ILiveTileModelProvider
{
    public LiveTileModel GetModel(LiveTileContext liveTileContext)
    {
        if (liveTileContext == null)
        {
            throw new ArgumentNullException("liveTileContext");
        }

        return CacheHelper.Cache(() =>
        {
            if (!ActivitySettingsHelper.OnlineMarketingEnabled(
                liveTileContext.SiteInfo.SiteName))
            {
                return null;
            }

            var newContactsCount = GetNewContactsCount();

            return new LiveTileModel
            {
                Value = newContactsCount,
                Description = "New contacts in the last 14 days",
            };
        }, new CacheSettings(2, "ContactManagementLiveTileModelProvider",
            liveTileContext.SiteInfo.SiteID));
    }

    /// <summary>
    /// Gets total number of new contacts from the last 14 days.
    /// </summary>
    /// <returns>Number of new contacts</returns>
    private static int GetNewContactsCount()
    {
        return ContactInfoProvider.GetContacts()
            .CreatedAfter(DateTime.Now.AddDays(-14))
            .Count;
    }
}
```

3. Register the live model provider via an assembly attribute using the *module code name* and *UI element code name* represented by the Live tile.

Find the *module code name* when editing the specific module in the **Modules** application. For Kentico modules, you can also use the constant defined in the *CMS.Core.ModuleName* class.

Find the *UI element code name* in **Modules -> UI elements** on the **General** tab of the UI element.



Registering a custom module

```
[assembly: RegisterLiveTileModelProvider("CustomModuleCodeName", "UIElement",  
typeof(ContactManagementLiveTileModelProvider))]
```

Registering a Kentico module

```
[assembly: RegisterLiveTileModelProvider(ModuleName.ABTEST, "ContactsFrameset",  
typeof(ContactManagementLiveTileModelProvider))]
```

Note that registering an existing live model provider overwrites the functionality. This way, you can change the behaviour of the default Live tiles.

```
[assembly: RegisterLiveTileModelProvider(ModuleName.ONLINEMARKETING,  
"ContactsFrameset", typeof(ContactManagementLiveTileModelProvider))]
```

4. Save the file.

The Live tile now displays the number of contacts added in the last 14 days.

