



In Kentico, you can modify how the system decides which [currency](#), [payment method](#) and [shipping option](#) are preferred for a registered [customer](#). The customer's preferred currency, payment method and shipping option are pre-filled when the customer goes through the [checkout process](#).

By default, the preferred currency, payment method and shipping option are chosen based on the customer's last order.

To customize how the system chooses the preferred currency, payment method and shipping option:

1. Open your Kentico project in Visual Studio.
2. Create a new class that implements the **ICustomerPreferencesProvider** interface (found in the *CMS.Ecommerce* namespace).



Class location

For production sites, we recommend creating a new assembly (Class library project) in your Kentico solution and including the classes there. Then, add the appropriate references to both the assembly and the main Kentico web project. For more information, see [Best practices for customization](#).

3. Implement the **GetPreferences** method.
 - The method must return an object of the **CustomerPreferences** class, which contains the *CurrencyID*, *PaymentOptionID* and *ShippingOptionID* properties (all nullable integers). Assign *null* values to the properties that you do not want to set.
4. Register your *ICustomerPreferencesProvider* implementation using the **RegisterImplementation** assembly attribute.

```
using CMS;
using CMS.Ecommerce;

[assembly: RegisterImplementation(typeof(ICustomerPreferencesProvider), typeof(
    CustomCustomerPreferencesProvider))]

public class CustomCustomerPreferencesProvider : ICustomerPreferencesProvider
{
    public CustomerPreferences GetPreferences(CustomerInfo customer,
        SiteInfoIdentifier site)
    {
        ...
    }
}
```

5. Save your project and reload your Kentico website.

If you open your website, the system uses the registered *ICustomerPreferencesProvider* implementation. Registered customers now have their currency, payment method and shipping option pre-filled based on your custom logic.