If you want to limit the availability of payment methods based on a specified shipping option, shipping country or a different attribute from the shopping cart, you need to customize the **PaymentOptionInfoProvider** class from the **CMS.Ecommerce** namespace.

The following example disables the *PayPal* payment method when the customer selects USA in the shipping address or selects *U SPS* as the shipping option.

> **Note**: The example assumes that the code name of the corresponding payment method is *PayPal*.

Start by prepare a separate project for custom classes in your Kentico solution:

1. Open your Kentico solution in Visual Studio.
2. Create a new *Class Library* project in the Kentico solution (or reuse an existing custom project).
3. Add references to the required Kentico libraries (DLLs) for the new project:
    a. Right-click the project and select **Add -> Reference**.
    b. Select the **Browse** tab of the **Reference manager** dialog, click **Browse** and navigate to the ***Lib*** folder of your Kentico web project.
    c. Add references to the following libraries (and any others that you may need in your custom code):
        - **CMS.Base.dll**
        - **CMS.Core.dll**
        - **CMS.DataEngine.dll**
        - **CMS.Ecommerce.dll**
        - **CMS.Globalization.dll**
        - **CMS.Helpers.dll**
4. Reference the custom project from the Kentico web project *(CMSApp* or *CMS)*.
5. Edit the custom project's **AssemblyInfo.cs** file (in the *Properties* folder).
6. Add the **AssemblyDiscoverable** assembly attribute:

```
using CMS;

[assembly:AssemblyDiscoverable]
```

Continue by creating a custom implementation of the *PaymentOptionInfoProvider* class:

1. Add a new class under the custom project, for example named **CustomPaymentOptionInfoProvider**.
2. Make the class inherit from the **PaymentOptionInfoProvider** class.
3. Register the custom provider class using the **RegisterCustomProvider** assembly attribute.
4. Override the provider's **IsPaymentOptionApplicableInternal** method:

```
using System;

using CMS;
using CMS.Ecommerce;
using CMS.Globalization;

[assembly: RegisterCustomProvider(typeof(CustomPaymentOptionInfoProvider))]

public class CustomPaymentOptionInfoProvider : PaymentOptionInfoProvider
{
    protected override bool IsPaymentOptionApplicableInternal(ShoppingCartInfo
cart, PaymentOptionInfo paymentOption)
    {
        // Does not check availability if the shopping cart or payment option is
not available
        if ((cart == null) || (paymentOption == null))
        {
            return true;
        }

        // Allows all payment methods except for PayPal
        if (!paymentOption.PaymentOptionName.Equals("PayPal", StringComparison.
InvariantCultureIgnoreCase))
        {
            return true;
        }

        // Disables PayPal if the billing address is within the USA
        if (cart.ShoppingCartBillingAddress != null)
        {
            CountryInfo country = CountryInfoProvider.GetCountryInfo(cart.
ShoppingCartBillingAddress.AddressCountryID);
            if ((country != null) &&
                country.CountryThreeLetterCode.Equals("USA", StringComparison.
InvariantCultureIgnoreCase))
            {
                return false;
            }
        }

        // Disables PayPal if the USPS shipping method is selected
        return (cart.ShippingOption == null) ||
            !cart.ShippingOption.ShippingOptionName.Equals("USPS",
StringComparison.InvariantCultureIgnoreCase);
    }
}
```

5. Save all changes and **Build** the custom project.

If a customer now selects USA as the country in their billing address or selects *USPS* as their shipping option, they will not be able to use *PayPal* as the payment method.

If you want to use the user interface for setting the limitations, you can either create a user interface or create a custom table and use the custom table API to add the values for the *Dictionary* variable with the limitations. Do not forget to set caching (you can use the CacheHelper class).