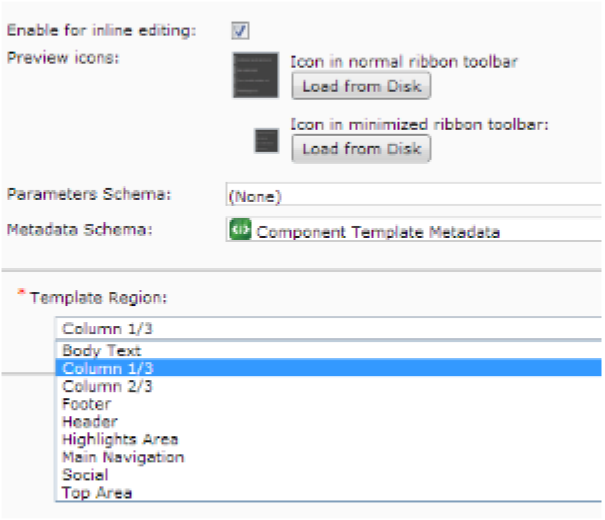
# Implementing Page Regions

To enable the proper usage of Experience Manager, we have decided to implement the concept of page regions, and tie Component Templates to a given region.

Using regions we can show on experience manager visually what areas you can drag and drop your components to. Also the region constrains regarding schema and component template pairs, allow us to change the presentation automatically(component template) when the component is dropped into a different region. So the same content can be rendered differently, depending on the region where it is placed.

This gives us a limitation that a given Component Template can only be used within a given Region, but this was considered to be an acceptable limitation.

You can tie a component to a Region by using the **Component Template Metadata** schema and selecting the keyword that defines your region.



Enable for inline editing: ☑

Preview icons: Icon in normal ribbon toolbar

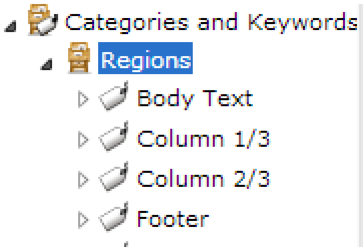
Icon in minimized ribbon toolbar:

Parameters Schema: (None)

Metadata Schema: Component Template Metadata

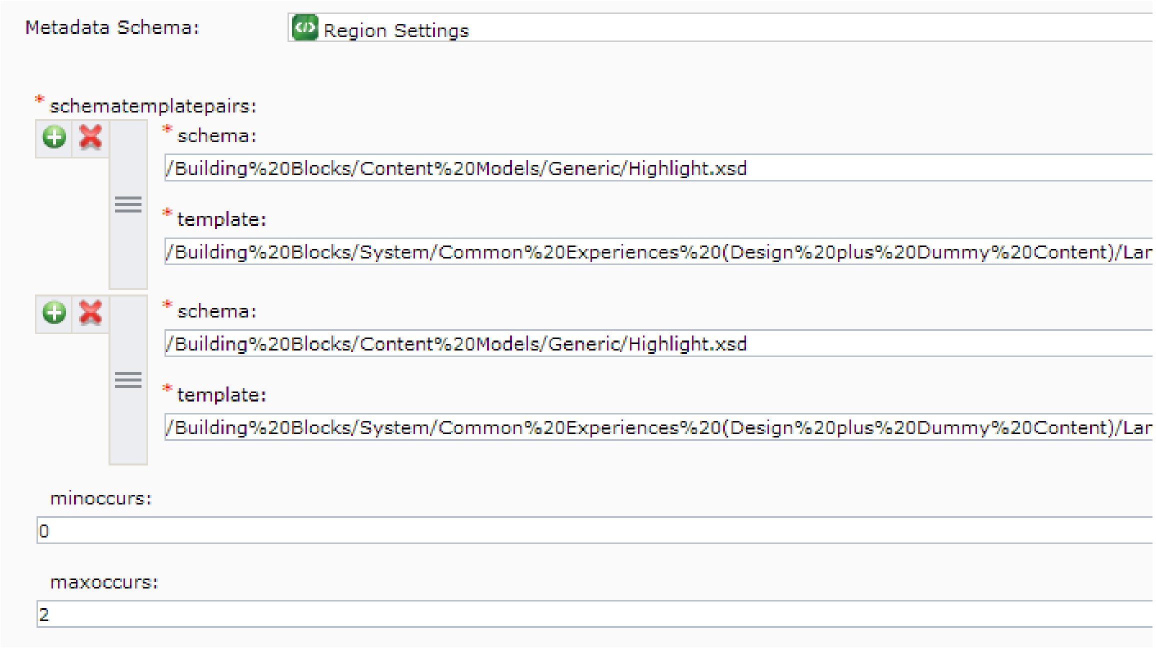
* Template Region: Column 1/3

## Defining a region



If you need to define a new region, you can simply add a keyword to the Regions category.

This region keyword must use the "Region Settings" metadata schema, and you must define the region constraints that apply to it. In particular, you **must** define the content types that are allowed within that region.



Metadata Schema: Region Settings

* schematemplatepairs:

* schema:
/Building%20Blocks/Content%20Models/Generic/Highlight.xsd

* template:
/Building%20Blocks/System/Common%20Experiences%20(Design%20plus%20Dummy%20Content)/Lar

* schema:
/Building%20Blocks/Content%20Models/Generic/Highlight.xsd

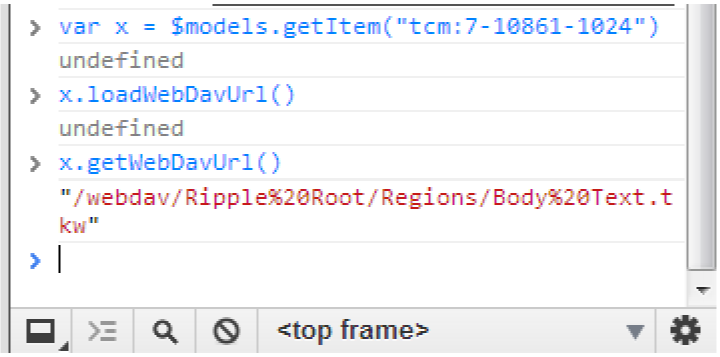
* template:
/Building%20Blocks/System/Common%20Experiences%20(Design%20plus%20Dummy%20Content)/Lar

minoccurs:
0

maxoccurs:
2

As can be seen above, the constraints have to be defined by using the schema and template's *relative webdav url* **without including /webdav/publication name**. You can easily find an item's webdav Url by using the javascript console in chrome or firefox and typing the following three commands:

```
var x = $models.getItem("tcm:1-2-3")
x.loadWebDavUrl()
x.getWebDavUrl()
```

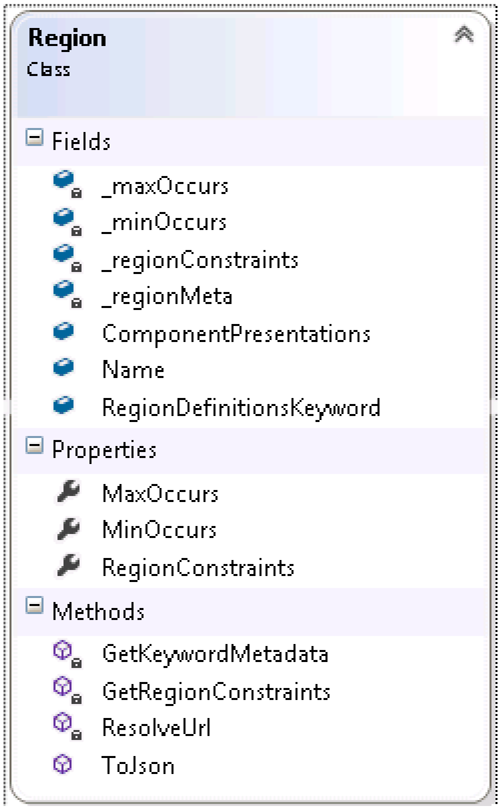


```
> var x = $models.getItem("tcm:7-10861-1024")
  undefined
> x.loadWebDavUrl()
  undefined
> x.getWebDavUrl()
  "/webdav/Ripple%20Root/Regions/Body%20Text.t
  kw"
> |
```

When outputting a Page you can use the page extension method page.*GetRegions()* to retrieve a list of Regions in the page.

```
public static Dictionary<string, Region> GetRegions(this Page page)
```

The Region object is described below:



**Region**
Class

Fields
- _maxOccurs
- _minOccurs
- _regionConstraints
- _regionMeta
- ComponentPresentations
- Name
- RegionDefinitionsKeyword

Properties
- MaxOccurs
- MinOccurs
- RegionConstraints

Methods
- GetKeywordMetadata
- GetRegionConstraints
- ResolveUrl
- ToJson

We have also developed a Dreamweaver custom function, "RenderRegion", that will render all component presentations within a given Region in a page, and output the region markup as required by Experience Manager.

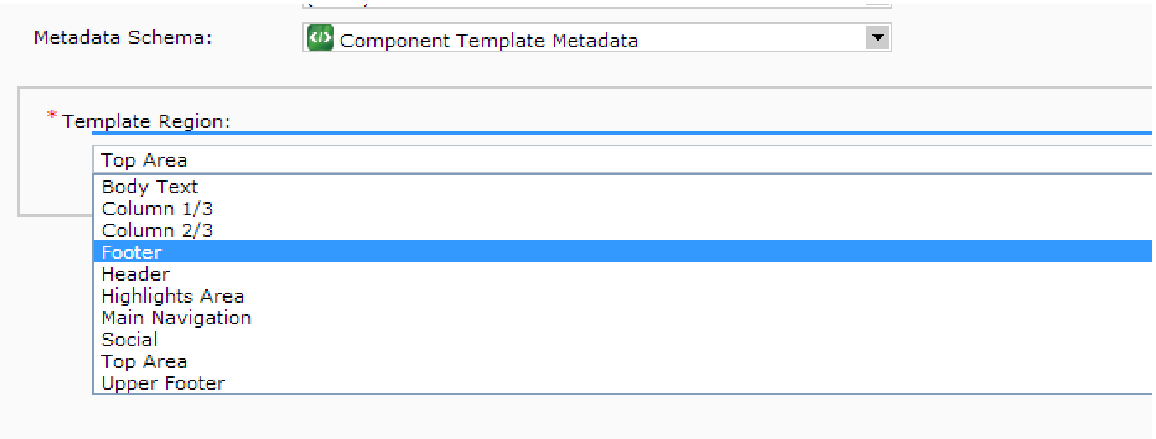
The Region Markup will only be outputted if the Publication Target is SiteEdit enabled.

Use this extension like this:

```
<div class="someclass">@@RenderRegion("Footer")@@</div>
```

## Marking Component Templates to belong a Region

Component Templates are tagged to belong a Region using the "**Component Template Metadata**".



Metadata Schema: Component Template Metadata

* Template Region:

Top Area

- Body Text
- Column 1/3
- Column 2/3
- Footer
- Header
- Highlights Area
- Main Navigation
- Social
- Top Area
- Upper Footer

As mentioned earlier this has the limitation that a given Component Template can only be used within a given Region on the same page. The workaround for this is to create a copy of the template. (Ángel: It is arguable how often we are actually going to use the same CT in different Regions of the page. From the templates developed during the POC, this only happened once with the Columns Titles. )

The benefits is that by limiting the Page Regions to only use one CT with each schema, we get the CT switched automatically when the component is moved in Experience Manager, and by extension the CP is region is also switched automatically.

## Extending a region

Obviously, different template themes will have different restrictions for what is essentially the same region. To cater for a scenario where, for instance, you could have a footer with a different number of components, we create the "Theme" category. Essentially, these are the steps required to override the "Footer" region for a theme named "Simple Theme".

1. Create a keyword in the "Themes" category and name it "Simple Theme". Do no use any metadata.
2. Create a child keyword of the "Simple Theme" keyword, and name it "Simple Theme - Footer".
    a. Use the Region Settings Metadata Schema and define the constraints you have for the Footer region
    b. Mark this child keyword as an **abstract keyword**
3. Open your Page Template and use the "Page Template Metadata" schema. Select "Simple Theme" as the theme for this page template.

The code in *RenderRegion* will find the correct region constraints to use. **It is extremely important that you observe the syntax for overriding regions in a theme. The code will try to find these region overrides by using a simple string compare on the keyword title, so be careful with CASE and SPACES.**