

Image Cropper Extended Dokumentation



Was ist der Image Cropper Extended

Der Image Cropper Extended basiert auf dem Umbraco eigenen Image Cropper und ist ein Datentyp, der es erlaubt eine Sammlung von Bildausschnitten für ein bestimmtes Upload Property zu definieren. Er kann auf drei verschiedene Arten verwendet werden:

1. Als Property des Image-Mediatypen (Settings/Einstellungen Bereich)
2. Als Property eines Dokumenttyps in Verbindung mit einem Upload-Property (Settings/Einstellungen Bereich)
3. Als Property eines Mitglieder-Typs in Verbindung mit einem Upload-Property (Members/Mitglieder Bereich)

Wie der Image Cropper Extended verwendet wird

Es können unterschiedliche Cropper-Datentypen für verschiedene Zwecke mit eigenen Datentyp-Einstellungen definiert werden. Zum Beispiel für Dokumenttypen, die mehr als ein Upload-Property haben. Jeder Cropper wird mit einem bestimmten Property über dessen Alias verbunden.

The screenshot displays the Umbraco administration interface for configuring an 'Image Cropper Extended' data type. The left-hand 'Developer' sidebar lists various data types, with 'Image Cropper Content 2' highlighted. The main content area, titled 'Edit datatype: Image Cropper Content 2', contains the following configuration fields:

- Name:** Image Cropper Content 2
- Property editor:** Image Cropper Extended
- Property editor GUID:** 5e8e0761-22b0-42db-8a99-dc7dd4ce797b
- Settings:**
 - Property alias:** teaserImageUpload
 - Save crop images as files:** ☒ Quality 90
 - Ignore ICC:** ☐ (If checked the ICC profile embedded in the source image will be ignored.)
 - Show ignore ICC in editor:** ☐ (If checked you can set the ignore ICC option for each image item.)
 - Show Label:** ☐
 - Crops:** A section for defining crop constraints, including fields for Name, Target width, Target height, Default position, and Keep aspect.

Datentyp-Einstellungen für einen Image Cropper Extended

Der Image-Mediatyp hat standardmäßig ein Upload-Property mit dem **umbracoFile** definiert. Weshalb dies die Grundeinstellung für neu erzeugte **Image Cropper Extended**-Datentypen ist.

Die Einstellungen des Image Cropper Extended im Detail

Property alias:

Über diese Eigenschaft wird der Cropper mit einer Upload-Property z. B. dem umbracoFile des Image-Medientyps.

Save crop images as files:

Sollen die definierten Bildausschnitte serverseitig als eigene Bilddateien erzeugt werden, dann haken Sie diese Option ab und definieren Sie die Qualität (0-100) der JPEG-Kompression. Andernfalls werden nur die Koordinaten des Auswahlrechtecks gespeichert.

Ignore ICC:

Evtl. vorhandene ICC-Profile werden ignoriert.

Show ignore ICC in editor:

Die ICC-Option wird für jedes Bild angezeigt und separat behandelt.

Show Label:

Zeigt den Property-Namen im Editor an, was allerdings die verfügbare Breite des Editor verringert.

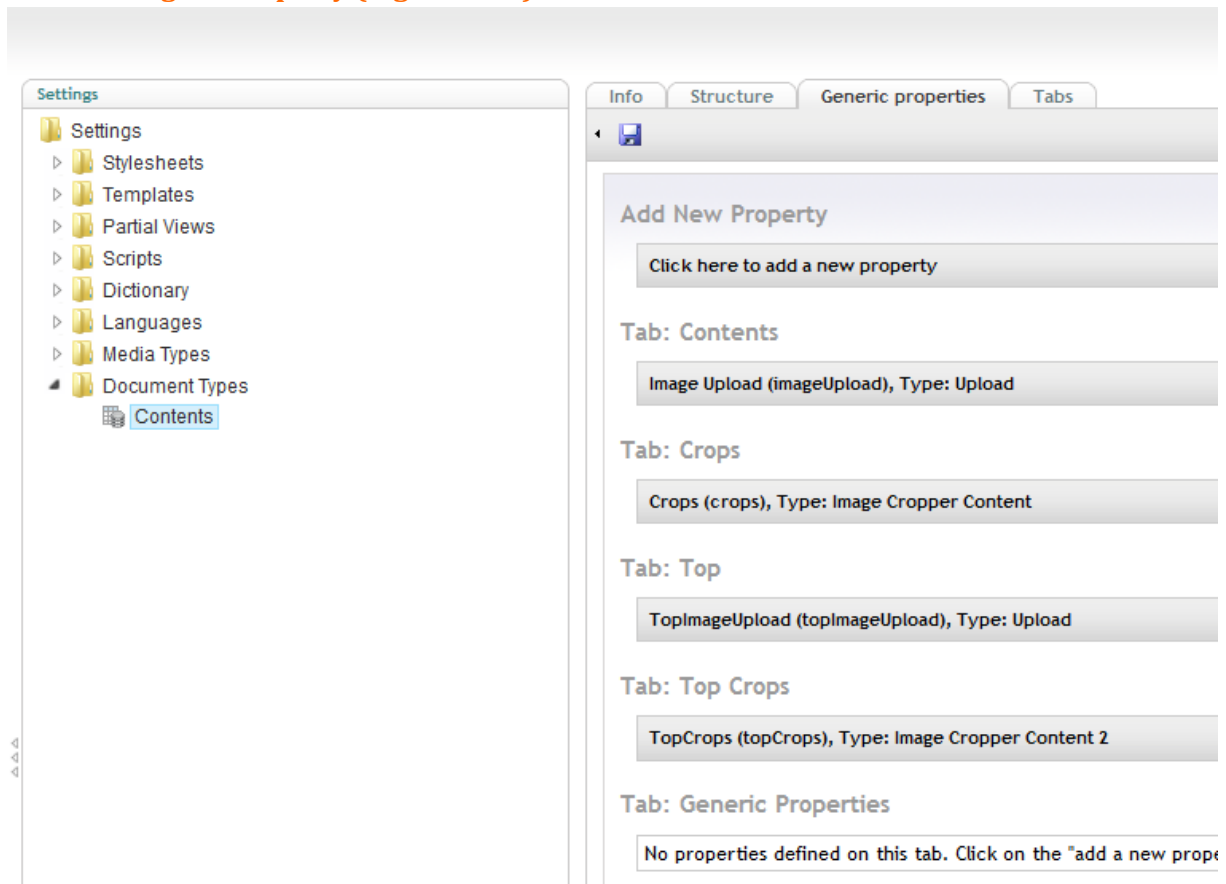
Crops:

Zuletzt werden die verschiedenen Ausschnittsdefinitionen festgelegt. Jeder Cropper kann mehrere Ausschnitte definieren. Über den Namen können die Eigenschaften (z. B. der URI des serverseitig erzeugten Bildes) eines Ausschnittes für die Ausgabe abgerufen werden. Target width und height bestimmen die Ziel-Dimensionen des Ausschnitts. Wird einer der beiden leer gelassen oder auf 0 gesetzt, kann die entsprechende Dimension beliebig gewählt werden. In diesem Fall werden die beiden folgenden Einstellungen ignoriert.

Default Position bestimmt die Startposition des Bildausschnitts und Keep aspect sorgt dafür, dass die ursprünglichen Seitenverhältnisse erhalten bleiben – der Ausschnitt einen unverzerrten Teil des Originals abbildet.

Nicht vergessen den Datentyp zu sichern! Das Hinzufügen eines weiteren Crops führt nicht zum Sichern des Datentyps.

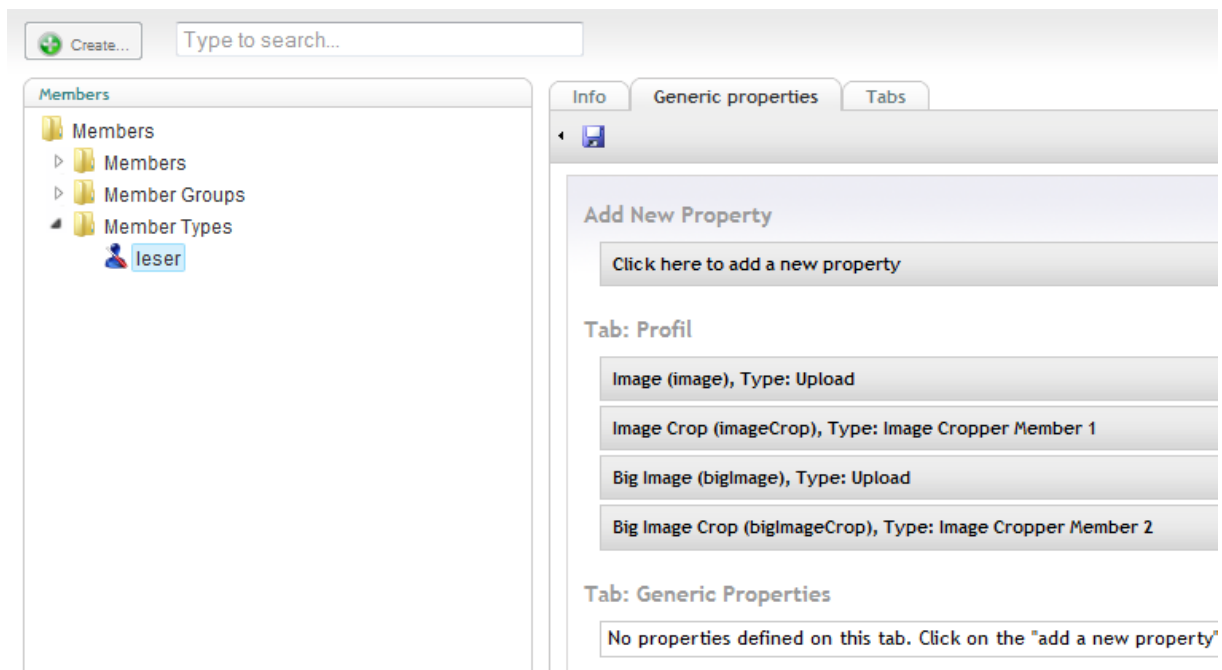
Verwendung als Property (Eigenschaft)



Als Eigenschaft eines Dokumenttyps



Als Eigenschaft des Image-Medientyps



Als Eigenschaft eines Mitglieder-Typs (analog Dokumenttyp)

Der Cropper als Eigenschaft eines Dokumenttyps

Der folgende Screenshot zeigt eine Cropper-Eigenschaft für den Image-Medientyp innerhalb des Medien-Bereichs im Einsatz.

Unterhalb des Bildes mit dem Auswahlrechteck sind die definierten Ausschnitte als Schaltflächen zu sehen. Über diese kann zwischen den verschiedenen Ausschnitten gewechselt werden. Sie zeigen auch an, ob der gewählte Ausschnitt zu einer Vergrößerung führt. Die Schaltfläche erscheint in rot. Sie müssen selbst entscheiden, ob Sie den Qualitätsverlust akzeptieren wollen, der Cropper führt die Vergrößerung jedenfalls aus.

Achtung!

Der Image Cropper Extended bietet für Dokumenttypen eine Preview-Unterstützung. Um diese zu nutzen verwenden Sie das neue Attribut **newurl**. Das Attribut **url** verhält sich wie beim Umbraco Image Cropper. Hier ein XSLT-Kodefragment:

```
...
<xsl:param name="currentPage"/>
<xsl:template match="/">
  <xsl:variable name="imageUrl"
    select="$currentPage/crops/crops/* [@name = 'teaser']/@newurl"/>
  <xsl:if test="$imageUrl != ''">
    <img style="float:left">
      <xsl:attribute name="src">
        <xsl:value-of select="$imageUrl" />
      </xsl:attribute>
    </img>
  </xsl:if>
</xsl:template>
...
```

crops ist der Alias-Name der Dokumenttyp-Eigenschaft vom Datentyp **Image Cropper Content** s.o.

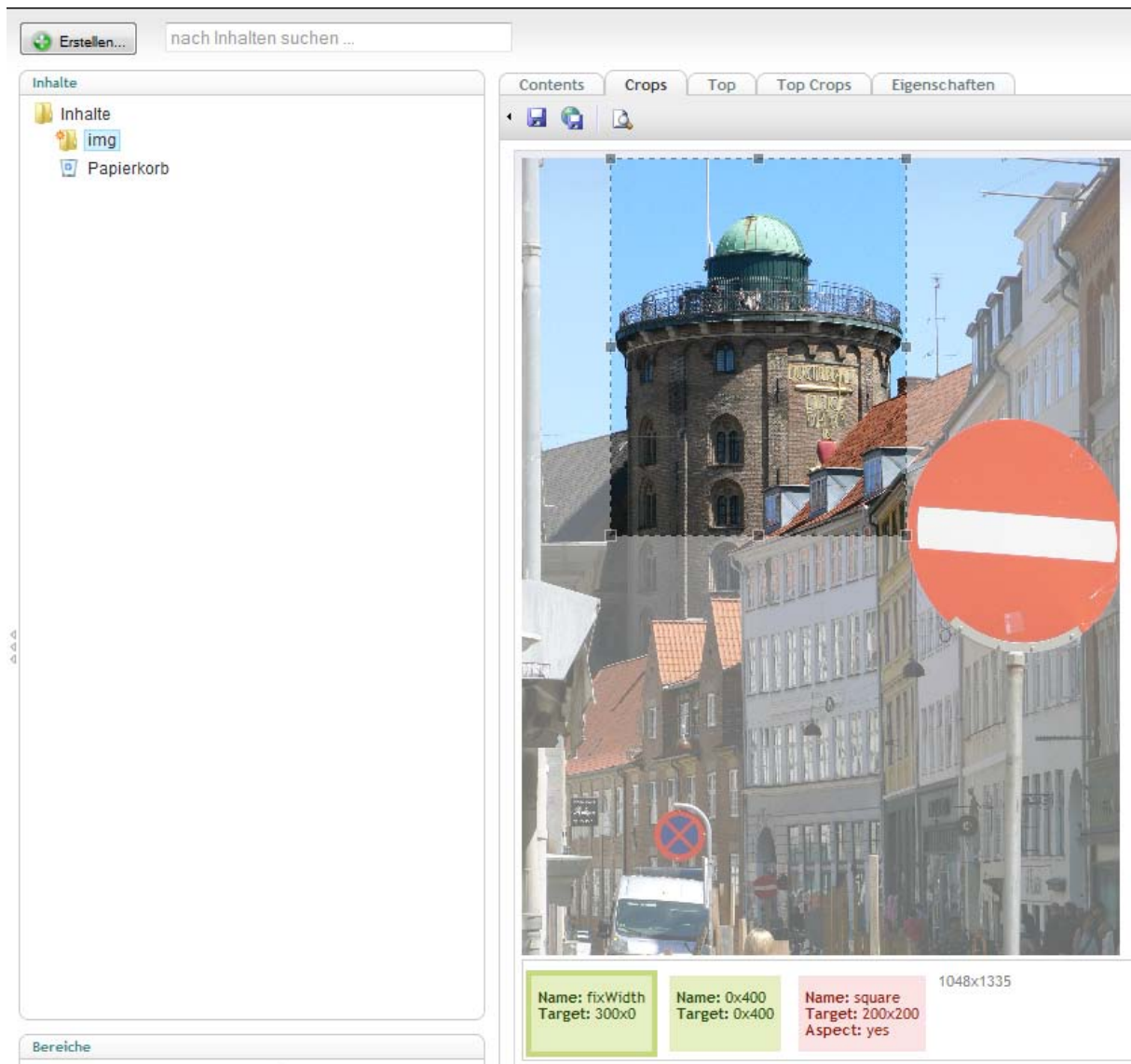


Image Cropper im Content-Bereich.

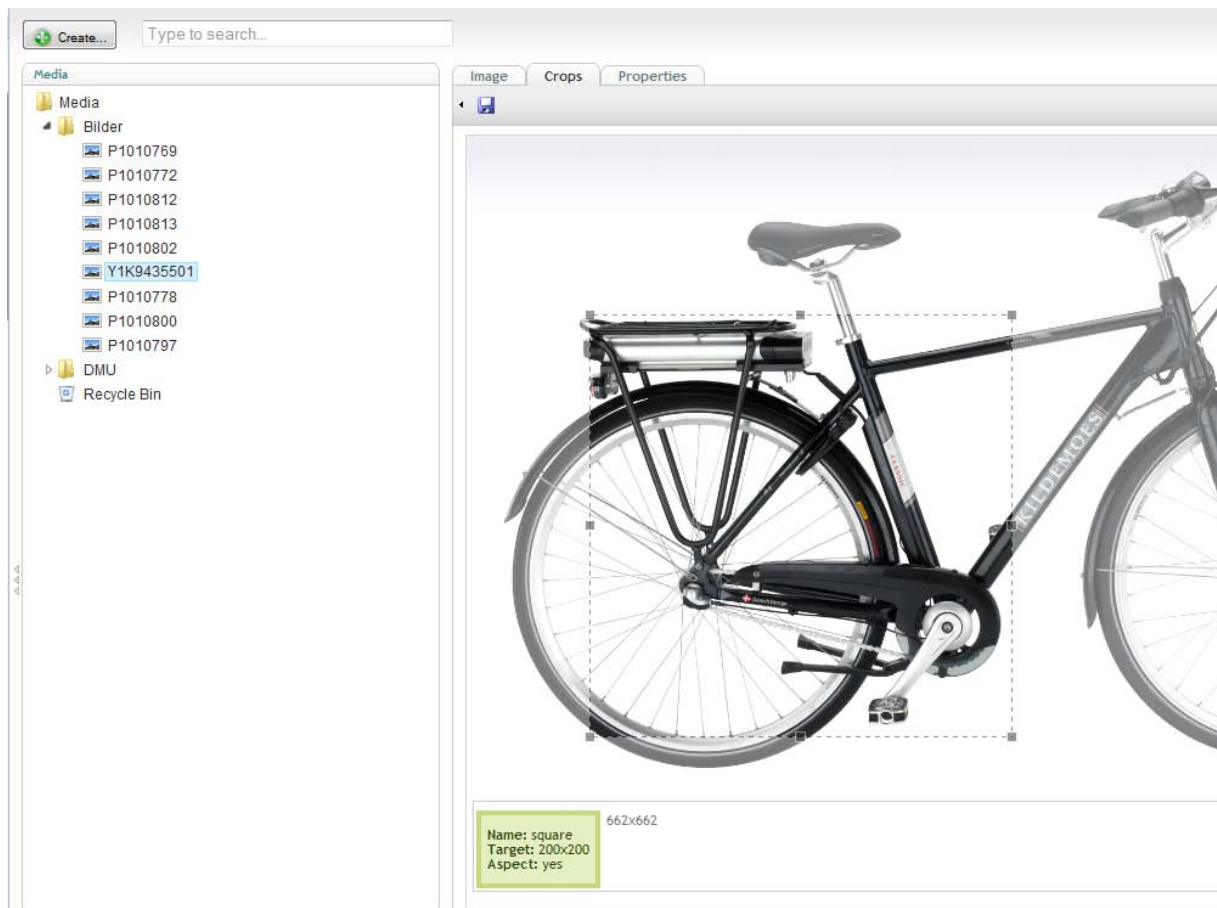
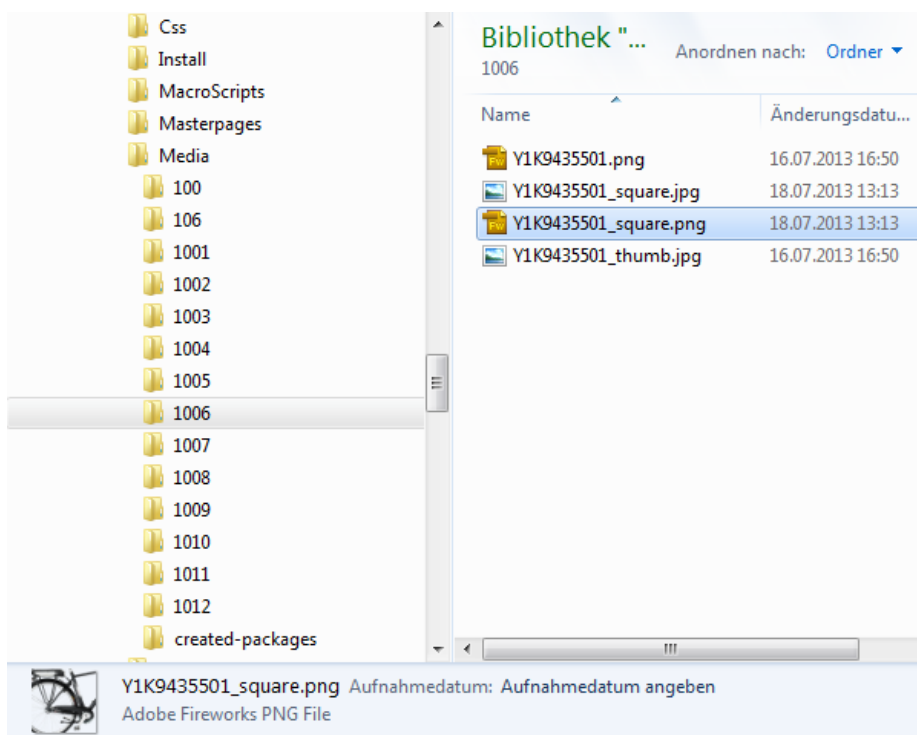


Image cropper im Medien-Bereich



Ansicht der erzeugten Bilddateien im Explorer (Windows7)

Razor

Folgendes Kodefragment zeigt das entsprechende Beispiel für ein Razor-Makro:

```
@inherits umbraco.MacroEngines.DynamicNodeContext
@if (Model.Crops != null)
{
    <strong>razor</strong><br />
    
}
```

MVC

Für die einfache Verwendung in MVC-Views ist ein **Property Editor Value Converter** (kurz: PEVC) definiert. Nachfolgend ein Beispiel für einen partial view:

```
@using idseefeld.de.imagecropper.PropertyEditorValueConverter;
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    var typesImageCropperExtended = Model.Content.GetPropertyValue<CropList>("topCrops");
}
@if (typesImageCropperExtended.Any())
{
    foreach (var crop in typesImageCropperExtended) {
        <h4>@crop.Name</h4>

         <br />
    }
}
@{
    // select a single crop by name
    var teaserCrop = typesImageCropperExtended.Find(n => n.Name.Equals("teaser"));
}
<h3>selected crop</h3>
<h4>teaser</h4>

```

Bis zur Umbraco Version 6.2.0 fehlt die volle Unterstützung von typed content für Media Objekte. In der Zwischenzeit können Sie Ihren Code auf folgende Weise einsetzen. Über einen Media Picker wird ein definierter Ausschnitt des gewählten Bildes angezeigt:

```
@using idseefeld.de.imagecropper.PropertyEditorValueConverter;
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    idseefeld.de.imagecropper.Model.CropModel crop = null;
    idseefeld.de.imagecropper.Model.ImageCropperModel crops = null;
    CropList cropList = null;
    var mediaItem = Umbraco.TypedMedia(Model.Content.GetPropertyValue<int>("myMediaPicker"));
    string cropValue = String.Empty;
    if (mediaItem != null) {
        //Image Cropper Extended property with alias name "crops" assigned to Media Type Image
        cropList = mediaItem.GetPropertyValue<CropList>("crops");
        cropValue = mediaItem.GetPropertyValue<string>("crops");
        //from v6.2.0 cropValue will return (Collection), what makes the following if clause necessary
        if (cropList == null && !String.IsNullOrEmpty(cropValue)) {
            crops = new idseefeld.de.imagecropper.Model.ImageCropperModel(cropValue);
        }
    }
    if (crops != null) {
        //Image Cropper Extended data type contains a crop definition with name "square"
        crop = crops.Find("square");
    }
    if (cropList != null) {
        crop = cropList.Find("square");
    }
}
<h3>Named Crop of Media Picker Image</h3>
@if (crop != null) {
    
}
}
```

Umbraco Version 6.2.0 wird *typed media properties* unterstützen, so dass sich der voranstehende Code folgendermaßen verkürzen lässt:

```
@using idseefeld.de.imagecropper.PropertyEditorValueConverter;
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    idseefeld.de.imagecropper.Model.CropModel crop = null;
    CropList cropList = null;
    var mediaItem = Umbraco.TypedMedia(Model.Content.GetPropertyValue<int>("myMediaPicker"));
    if(mediaItem!=null){
        //Image Cropper Extended property with alias name "crops" assigned to Media Type Image
        cropList = mediaItem.GetPropertyValue<CropList>("crops");
    }
    if(cropList!=null){
        //Image Cropper Extended data type defines a crop named: square
        crop = cropList.Find("square");
    }
}
<h3>Named Crop of Media Picker Image</h3>
@if (crop != null)
{
    
}
```

Neuer ImageProvider "imageCropperProvider"

Für das neue Umbraco Server Control für Bilder ist ein ImageProvider implementiert. Dieser entspricht im wesentlichen dem Standard-Provider, unterstützt jedoch PNG Bilder da der Image Cropper Extended die korrekte Erweiterung sichert. Auf einer Masterpage wird das Control wie folgt verwendet:

```
<umbraco:Image runat="server"
    field="theImage"
    Parameters="crop=square"
    Provider="imageCropperExtended" />
```

theImage ist der Alias-Name einer Eigenschaft vom Typ media picker.

Weitere Details beschreibt Morten Bock Sørensen auf seinem Blog:

[Introducing The umbraco:image Control](#)

ImageResizer settings

Die Bilderzeugung basiert auf dem freien Teil der ImageResizer.dll. Weitere Informationen finden sich auf: <http://imageresizing.net/>

Standardmäßig wird dadurch die Größe eines Bildausschnitts auf 3200x3200 Pixel beschränkt. Was in der Regel ausreichen sollte. Werden jedoch größere Ausschnitte benötigt genügen folgende zusätzliche Zeilen in der web.config:

```
<configuration>

    <configSections>
        ...

        <section name="resizer" type="ImageResizer.ResizerSection,ImageResizer"
requirePermission="false" />
    </configSections>
    <resizer>
        <sizelimits imageWidth="0" imageHeight="0" totalWidth="3200"
totalHeight="3200" totalBehavior="throwexception" />
    </resizer>
    ...
</configuration>
```

Einfach totalWidth und totalHeight wie benötigt anpassen. Mehr Details sind auf zu <http://imageresizing.net/docs/configuration-all> lesen.

Project Links

Der Quellcode befindet sich auf GitHub:

<https://github.com/idseefeld/imagecropper4umbraco>

Fehlerberichte bitte auf <https://github.com/idseefeld/imagecropper4umbraco/issues/> einstellen.

Umbraco package download:

<http://our.umbraco.org/projects/backoffice-extensions/image-cropper-extended>