Adding a new Text based property to Translation Manager

Out of the box Translation Manager will recognise the majority of text based properties within Umbraco, but if you have a custom property or you have installed a package that translation manager doesn't recognise you can extend the list of text properties it understands by adding them to the `translations.config` file within your site's `Config` folder.

```
<mappers>
     <text>
       <text>My.BoxedContent,My.SampleContent</text>
       <html>My.BoxedContent</html>
     </text>
</mappers>
```

In this example to property editors `My.BoxedContent` and `My.SampleContent` both contain text based values, and will be interpreted as text by Translation Manager, here `My.BoxedContent` also contains html (i.e is a rich text control) so translation manager will use the config to treat the text as html during translation.

Adding complex DataTypes to translation manager

If you have a property editor that stores its data in a more complex way you can still extend Translation manager to parse and recognise the text within the data. For this there are two options

1: Using the Custom JSON Mapper

If your property editor stores it's values within a JSON object - you can use the config file to tell translation manager how to interpret your properties data.

If for example you have a property that stores its data in the following format

```
{
     "Key", "some-key-value",
     "Title", "my property title",
     "Content", "<p>some rich text content</p>"
}
```

Then you could use the config options (below) to define the `Title` and `Content`

fields as text based.

Within the mappers section of the config file you can define this using the `<grid>` config (*n.b the control does not need to be within the grid to use this control*)

```
<mappers>
 <grid>
  <custom>
   <config alias="CustomProperty">
    <properties>
     <property name="Title" alias="Umbraco.Textbox" />
     <property name="Content" alias="Umbraco.TinyMCEv3" />
    </properties>
   </config>
  </custom>
 </grid>
</mappers>
```

This config tells translation manager to treat all `CutomProperty` values as JSON and extract the title and content values when creating and importing translations.

2: Write a custom ValueMapper

You can create your own custom mapper by Implementing the `IValueMapper` interface - there are a some examples online on how to do this :

- Example Stacked Content Value Mapper (Included in Translation Manager)
- Example for RJP.MultiURL Picker

A Note on Property Editor Naming

For the most part when looking for property editors, Translation Manager will use the Property Editor Alias set within the property editor - this is for example Umbraco.TextString for textboxes, and Jumoo.StyledText for the Styled Textbox property editor.

When you have a custom property editor you should use your property editor alias (alias in the package.manifest file) as the value in the configuration.

When looking for custom editors within the Grid property editor, translation manager will the folder the view is stored in, so for example if your grid editor contains the following config :

[config]

The editor alias for translation manager will be [alias] - which is the folder name, this is the name you should use for all configuration and custom value mappers.