

# Render Details

Each output format allows different rendering parameters to be set. In Elixir Report, these parameters are called render details. When you set the values using the Render Wizard, the values are remembered next time you save the report. Subsequent uses of the Render Wizard will restore those values.

For example, if you want PDF output to render as an image, you only have to tick the checkbox once. This value is now stored in the report (assuming you saved), so when anyone renders this report, from the designer, runtime or server, this stored value will be used.

Hence, the easiest way to configure render details for each output format is to configure them using the Render Wizard.

Sometimes you might want to vary render details through the API. To do this, you need to work with the `RenderDetails` object:

```
// assuming import com.elixirtech.report2.raw.model.RenderDetails;

RenderDetails rd = report.getRenderDetails("application/pdf"); // choose mime-type here

if (rd==null) // no render details for this mime-type stored
{

    // create a RenderDetails for the right type and add it to the report

    rd = new RenderDetails();

    rd.setMimeType("application/pdf");

    report.addRenderDetails(rd);

}

rd.setParameterValue("RenderAsImage","Yes");
```

This example sets the *RenderAsImage* flag to Yes for PDF output. The parameter value must always be a String and the value "Yes" (case-sensitive) is used to indicate a ticked checkbox.

Similarly, to set the Snap flag for Excel output, you would substitute mime-type *application/vnd.ms-excel* and use `rd.setParameterValue("Snap","Yes");`

To render a component based on the condition of the report format selected, you can use the above script to validate with the mimetype.

## **Code Snippet**

```
Renderer.getMimeType()=="application/vnd.ms-excel"
```

```
//Renderer.getMimeType(); will return the mimetype selected for report generation.
```

The above is only available for version 5.2 and above

Download the sample here: [Render Details Sample](#)