Custom Renderers

Including

- Introduction: Content at a glance.
- FrxTitle: Replace the report's page title and tab title by token replaced values.
- FrxMyReports: Display the user's list of reports, optionally limited to a single category.
- FrxSource: Display markup as a code snippet, without token replacement.
- FrxXML: Display the XML data source of the current data context, useful for debugging also.
- FrxParameterForm: Customize various aspects of the rendering of the report parameters input form.
- FrxSVGGraph: Render a graph (chart) in SVG format, using the PHP SVGGraph library.
- FrxInclude: Include another report as an asset with the appropriate tag, such as external SVG assets.
- FrxCrosstab: Display of a table in crosstab format.

Introduction

A renderer specifies how a tag in the report template is to render. A good example is to render a chart using the PHP SVGGraph library, as in the State Summary sample report:

```
<html id="frxsrc-1">
<head>
...
</head>
<body>
  ...
  <div frx:block="sampledb/users_by_state" id="users_by_state-block"
class="FrxSVGGraph">
    <svg id="state-chart" frx:renderer="FrxSVGGraph" frx:type="bargraph"
frx:xpath="*[total&gt;10000]" frx:color="{color}"
frx:link="sample.user_distribution_simple?state={state}"
frx:series_1="{total}" frx:label="{state}">
      </svg>
  </div>
  ...
</body>
</html>
```

FrxTitle

Use the content of this tag to override the title of the report. This allows both the page title and the tab title to be replaced by token replaced values in the report.

As an illustration of how to use this renderer, consider the User Distribution sample report:

```
<div id="frxsrc-2">
```

```
  <h2 frx:renderer="FrxTitle" id="frx-frxtitle">Users in cities in state
{name}</h2>
</div>
```

Because of the attributes we added to the **h2** HTML tag here (any HTML tag will do) the existing "2. Report that filters based on state lookup" title will be ignored.

Moreover, because of the **name** token included in this h2 tag, the rendered title will be dynamic.

For another illustration of this renderer, checkout the video about Dynamic page titles in reports.

FrxMyReports

Displays the user's list of reports (excluding hidden reports), optionally limited to a single category, as in this example:

No Reports Found

The above sample shows a list of all non-hidden reports in category **Sample**, using the category as the header generated by the following code:

```
<div id="frxsrc-3">
  <div frx:renderer="FrxMyReports" frx:category="Sample" id="frxmyrpts"/>
</div>
```

The following attributes are supported for the FrxMyReports renderer:

frx:category	Limit the list of reports to be shown to a particular category. Multiple categories may be used by specifying category_2 and category_3 attributes accordingly.

FrxSource

The FrxSource renderer displays markup as a code snippet. No token replacement is done for the children and all embedded code is escaped HTML. This is used in the Reporting Documentation to display the source of reports.

This renderer is used throughout the documentation as follows:

```
<div id="frxsrc-4">
  <div>
    <p>Embedded XHTML that you want displayed as source including {tokens}
that you want to display without being replaced</p>
  </div>
</div>
```

FrxXML

Displays the XML of the current data context, which is particularly useful for debugging purposes. If you embed this in a report, it will show you the XML data source that is used for token replacement, so it can give you a good idea as to what data is being returned and which tokens can be used.

Here is a sample that is displayed using the FrxXML renderer which shows the data returned by the datablock:

```
<book>
  <booktitle>WYSIWYG Reporting</booktitle>
  <bookfolder>help.reportingwysiwyg</bookfolder>
  <chapters>
    <chapter>
      <title>Introduction</title>
      <subtitle>Content at a glance</subtitle>
      <abstract/>
      <link>intro</link>
    </chapter>
    <chapter>
      <title>Create A New Report</title>
      <subtitle>Steps to start the creation of a new report</subtitle>
      <abstract/>
      <link>create</link>
    </chapter>
    <chapter>
      <title>Manage Data Blocks</title>
      <subtitle>Add, edit or remove data blocks in a report</subtitle>
      <abstract/>
      <link>datablocks</link>
    </chapter>
    <chapter>
      <title>General Report Options</title>
      <subtitle>Specify general options of a report</subtitle>
      <abstract/>
      <link>general</link>
    </chapter>
    <chapter>
      <title>Document Types</title>
      <subtitle>Select enabled document formats for optionally saving a
report</subtitle>
      <abstract/>
      <link>doctypes</link>
    </chapter>
    <chapter>
      <title>Report Layout Options</title>
      <subtitle>Specify various layout options for a report</subtitle>
      <abstract/>
      <link>layout</link>
    </chapter>
    <chapter>
      <title>Report Parameters</title>
      <subtitle>Specify various parameter options for a report</subtitle>
      <abstract/>
      <link>parameters</link>
    </chapter>
    <chapter>
```

```
      <title>Configure Report Fields</title>
      <subtitle>Define special formatting rules for any report
field</subtitle>
      <abstract/>
      <link>fields</link>
    </chapter>
  </chapters>
</book>
```

Be aware however that this FrxXML renderers removes XML comment lines (e.g. those used to specify data block security in XML format).

The above example was generated with the following code:

```
<div id="frxsrc-5">
  <div frx:renderer="FrxXML" id="frx-frxxml"
frx:block="forena_help/reportingwysiwyg_topics"/>
</div>
```

FrxParamterForm

Customize the standard report parameters input form. Use it on a div tag anywhere within the **body** part of your report to control various aspects of the rendering of the parameter form, as in the Roles sample report, for which the .frx file includes these lines:

```
<html id="frxsrc-6">
<head>
...
<frx:parameters>
  <frx:parm id="role" data_source="drupal/roles" type="select">3</frx:parm>
</frx:parameters>
...
</head>
<body>
  ...
  <div frx:renderer="FrxParameterForm" frx:title="Report Execution
Parameters" frx:collapsible="1" frx:collapsed="0" frx:submit="Show users and
their permissions" id="parmeter-form">
    <p>Role Description: {role}</p>
    <p>Select a role and hit the button to run the report.</p>
    <p>{submit}</p>
  </div>
  ...
</body>
</html>
```

The above example illustrates the following FRX attributes supported by the FrxParameterForm renderer:

frx:title	The title of the parameters field set.
frx:collapsible	Indicate if the form should be collapsible or not: • set to "1" to make the parameter form collapsible. • set to "0" for a parameter form that cannot be collapsed.
frx:collapsed	Indicate how a collapsible parameter form should be shown: • set to "1" for a collapsed form. • set to "0" for a not collapsed form. The default behavior is to expand the form only if no data was returned by the report.
frx:submit	The label of the submit button.

The children of the FrxParameterForm div allow you to specify the exact layout of the parameters form using Forena's token replacement syntax, which is illustrated in the above example via the content of the 3 paragraphs contained in the FrxParameterForm div. The default context is changed to be the rendered parameter form, so that the parameter ids will allow replacement of a form control.

Note: The parameter form is always rendered at the top (even if you would move it after the data blocks to be rendered).

FrxSVGGraph

Render a graph using the SVG format, using the PHP SVGGraph library. Make sure to install this library prior to using this renderer.

We'll use the State Summary sample report to illustrate how to use this renderer, for which the .frx file includes these lines:

```
<div id="frxsrc-7">
  <div frx:block="sampledb/users_by_state" id="users_by_state-block"
class="FrxSVGGraph">
    <svg id="state-chart" frx:renderer="FrxSVGGraph" frx:type="bargraph"
frx:xpath="*[total&gt;10000]" frx:color="{color}"
frx:link="sample.user_distribution_simple?state={state}"
frx:series_1="{total}" frx:label="{state}">
      </svg>
  </div>
</div>
```

The following attributes are supported for the FrxSVGGraph renderer:

frx:type	The type of graph to be rendered. If omitted, then **BarGraph** is assumed. These are the currently supported types of graphs: • BarGraph • Bar3DGraph • StackedBarGraph • GroupedBarGraph • GroupedBarGraph • CylinderGraph • StackedCylinderGraph • GroupedCylinderGraph • PieGraph • Pie3DGraph • HorizontalBarGraph • LineGraph • MultiLineGraph • ScatterGrap • MultiScatterGraph • RadarGraph • MultiRadarGraph
frx:xpath	The XPATH expression for the data to be graphed (e.g. ***[total>10000]**, which is what is used in the sample (note that only states are included in the graph that have a total above 10000). If omitted, then an XPATH expression of * is assumed and all data is graphed.
frx:link	Create a hyperlink for the data to be graphed (e.g. **sample.user_distribution_simple?state={State}**. Use field tokens to generate the link dynamically.
frx:series	The column containing the series of the graph. Multiple series may be specified using an attribute of **frx:series_1** for the first series, **frx:series_2** as the second and so on.
frx:label	The label that should be used for the series. Usually this is specified using tokens (e.g. **{State}** in our sample).
frx:wrap_label	When specified, graph labels are word wrapped at the indicated number of characters using the php wordwrap function.
frx:options	This legacy attribute is currently still supported, but it is recommended to replace them by the corresponding **frx:xyz** attributes. Here is a sample of how it was used: **frx:options="series[]={total}&label={State}&colors[]=green&colors[]=yellow"**

In addition to the attributes mentioned above, any attribute supported as PHP SVGGraph options may be included as **frx** attributes also. Here are a few examples of some often used attributes:

frx:color	Specify a graph color (e.g. **frx:color="{color}"**). To specify multiple graph colors, you can specify **frx:color_1="red"** and **frx:color_2="blue"** also.
frx:width	Width of the graph to be rendered (e.g. **frx:width="720"**). If omitted, then **600** is assumed.
frx:height	Width of the graph to be rendered (e.g. **frx:height="480"**). If omitted, then **400** is assumed.

Checkout the PHP SVGGraph library documentation to fully understand the available options, as in this example:

```
<div id="frxsrc-8">
  <div frx:block="sampledb/users_by_state" id="users_by_state-block"
class="FrxSVGGraph">
    <svg id="state-chart" frx:renderer="FrxSVGGraph" frx:type="bargraph"
```

```
frx:xpath="*[total&gt;10000]" frx:color="{color}"
frx:link="sample.user_distribution_simple?state={state}"
frx:series_1="{total}" frx:label="{state}" frx:height="300" frx:width="450"
frx:axis_min_h="12500" frx:grid_division_h="500" frx:division_size_h="10"
frx:division_size_v="0" frx:show_subdivisions="true" frx:subdivision_size="5"
frx:graph_title_colour="red" frx:graph_title="Enhanced State Graph"
frx:graph_title_position="top" frx:graph_title_font_weight="bold"
frx:graph_title_font="georgia" frx:graph_title_font_size="18"
frx:label_h="Total nr of users" frx:label_v="State" frx:label_font="georgia"
frx:label_font_size="14" frx:label_colour="blue" frx:axis_text_angle_h="-60"
frx:back_colour="white">
      </svg>
  </div>
</div>
```

FrxInclude

This renderer facilitates report reuse within another report according to the "mode

include	Includes the report inline within the existing report, using the context of the current report as parameters to the included report.
ajax	Load the report in the current location via ajax. Note that the renderer must be provided a unique "id" attribute in order to properly load the link. This is an experimental setting at this time.
reference	Includes another report as an asset with the appropriate tag. The primary use of this mode is to create references to external SVG assets. When rendered in a web page, these need to be wrapped in embed tags, but when being included in a PDF document the raw SVG should be included. When used with normal reports it will embed a link to the report.

The following illustrates some common examples of using the FrxInclude renderer.

```
<div id="frxsrc-9">
    <div frx:src="reports/sample.state_graph.svg?height=400&amp;width=600"
frx:renderer="FrxInclude" frx:mode="reference" frx:height="480"
frx:width="640">...</div>
    <div frx:renderer="FrxInclude" frx:src="reports/sample.states"
frx:mode="include"/>
    <div frx:renderer="FrxInclude" frx:src="reports/sample.states"
frx:mode="ajax"/>
  </div>
```

FrxCrosstab

A **crosstab** table is designed to pivot data, that is to creat a table whose columns are determined by the values in a select statement/data block. The number of columns in a crosstab query is not constant but rather determined by the data returned by the block. It is typically used to transform data that looks like:

WA Female 100

WA	Male	50
ID	Female	25
ID	Male	30

Into a table that looks like:

	Female	Male
WA	100	50
ID	25	30

In order to do this you will need to specify a **grouping** attribute and a **dimension** attribute. The grouping attribute determines how to group the rows of data while the dimension attribute specifies which field will be used to create the columns out of.

The layout of the cross tab is follows:

```
<div id="frxsrc-10">
  <div id="watchdog_stats_block" class="FrxCrosstab"
frx:block="drupal/watchdog_stats">
    <table frx:renderer="FrxCrosstab" frx:group="{severity}{name}"
frx:dim="Msg Type: {type}">
      <thead>
        <tr>
          <th>Message</th>
          <th>Severity</th>
          <th>User Name</th>
          <td>Nr of msgs</td>
        </tr>
      </thead>
      <tbody>
        <tr id="watchdog_stats">
          <th>{message}</th>
          <th>{severity}</th>
          <th>{name}</th>
          <td>{typecount}</td>
        </tr>
      </tbody>
    </table>
  </div>
</div>
```

In the above layout the **td** elements in the **thead** section of the table tell the crosstab renderer to use that field as the dynamic columns that make up the pivot. The **frx:dim** attribute determines the values that will be used as columns in the table. Columns that are indicated with th elements are fixed. The structure is then mirrored in the **tbody** section. The **frx:group** attribute dictates what will be used to uniquely group the rows. One row will be generated for each unique frx:group specified.

FrxAjax

The FrxAjax renderer is used to place conditional Ajax Commands throughout the report. Commands that are intended to always be issued should instead by stored in frx:commands elements in the head, but in rare cases where a command should only be invoked under certain circumstances, they can be embeded in the report using this renderer.

Example:

```
<div>
    <script frx:renderer="FrxAjax" frx:command="invoke"
frx:selector="input#myinput" frx:method="attr"> ["checked", "checked"]
</script>
  </div>
```