

# **Geomajas printing plugin**

**Geomajas Developers and Geosparc**

---

# **Geomajas printing plugin**

by Geomajas Developers and Geosparc

1.18.3

Copyright © 2010-2012 Geosparc nv

---

---

# Table of Contents

1. Introduction .....	1
1. Back-end .....	1
2. Configuration .....	3
1. Dependencies .....	3

---

# List of Figures

1.1. Print components ..... 1

---

## List of Examples

2.1. Plugin dependency .....	3
------------------------------	---

---

# Chapter 1. Introduction

This printing plugin provides a simple yet compelling example of how platform independent web map printing might be implemented. Our implementation bypasses the browsers native printing functionality by rendering server side to a PDF document, which is then automatically downloaded to the browser. The PDF document can either be saved to the file system or opened in a new browser window from which native PDF printing is possible. As different browsers are allowed to - and will - render identical HTML pages in a slightly different manner, an absolute format like PDF is a necessity for creating reproducible prints.

Our implementation makes extensive use of the excellent iText library<sup>1</sup>, which allows direct drawing operations on a graphics context. This has the advantage of preserving the vectorial nature of the features in a layer, allowing crisp vector rendering at any zoom level.

## 1. Back-end

The printing plugin consists of a client part (the client GWT) that is responsible for the user interaction and a back-end part that is responsible for the generation of the PDF document. To achieve sufficient flexibility in the layout and composition of the PDF document, the document is conceptually a tree of nested components, each of which is responsible for its own rendering.

The following set of components have been implemented:

- *PageComponent*: a container component for the complete PDF page.
- *MapComponent*: this represents the main map. The layers, scale bar, legend and other components are child members of the map.
- *LabelComponent*: can be used for an arbitrary text label, e.g. a title.
- *ImageComponent*: represents an arbitrary static image, e.g. a north arrow on a map.
- *VectorlayerComponent*: draws a vector layer on the map.
- *RasterLayerComponent*: draws a raster layer on the map.
- *ScaleBarComponent*: draws a scale bar on the map.
- *LegendComponent*: draws a legend on the map.
- *ViewPortComponent*: a view port is a smaller map that is an inset of the main map and is used to highlight details of the main map.

### Figure 1.1. Print components

The layout is determined by recursively calculating the preferred size of each component first (which can vary according to font sizes, margins, etc...) followed by a second pass in which the layout constraints of each component are used to align the component with respect to its parent. This approach is top-down, starting with the root PageComponent. The following layout constraints are possible:

- *width*: the preferred width of the component.
- *height*: the preferred height of the component.
- *marginX*: the margin in the x-direction or the x-position in case of absolute alignment in the x-direction.

---

<sup>1</sup>iText is a library that allows you to generate PDF files on the fly", see <http://itextpdf.com/>

- *marginY*: the margin in the y-direction or the y-position in case of absolute alignment in the y-direction.
- *alignmentX*: the alignment type in the x-direction (left, right, justified or absolute).
- *alignmentY*: the alignment type in the y-direction (left, right, justified or absolute).

---

# Chapter 2. Configuration

Apart from the maven dependency configuration, most of the configuration options of the printing plugin are currently provided by the end user at runtime. To provide access to the plugin's functionality from the client GWT, an action has been provided that pops up a printing preferences window.

## 1. Dependencies

Make sure you include the correct version of the plugin in your project. Use the following excerpt (with the correct version) in the dependencyManagement section of your project:

```
<dependency>
  <groupId>org.geomajas.project</groupId>
  <artifactId>geomajas-project-server</artifactId>
  <version>1.18.3</version>
  <type>pom</type>
  <scope>import</scope>
</dependency>
```

If you are using geomajas-dep, this includes the latest released version of the caching plugin (at the time of publishing of that version). If you want to overwrite the caching plugin version, make sure to include this excerpt *before* the geomajas-dep dependency.

You can now include the actual dependency without explicit version.

### Example 2.1. Plugin dependency

```
<dependency>
  <groupId>org.geomajas.plugin</groupId>
  <artifactId>geomajas-plugin-print</artifactId>
</dependency>
```