

# **Geomajas geomajas-gwt2-widget plug-in guide**

**Geomajas Developers and Geosparc**

---

# **Geomajas geomajas-gwt2-widget plug-in guide**

by Geomajas Developers and Geosparc

2.1.1

Copyright © 2011 Geosparc nv

---

---

# Table of Contents

1. Introduction .....	1
2. Configuration .....	2
1. Dependencies .....	2
3. How-to .....	3
1. Replacing the default style through GWT .....	3
2. Replacing the default style through css .....	3
3. Replacing the complete view of one or more widgets .....	3
4. Usage examples .....	4

---

## List of Examples

2.1. Plug-in dependency .....	2
-------------------------------	---

---

# Chapter 1. Introduction

The core widget plugin contains a set of widely used widgets in a Geomajas application, as well as some very popular map controllers and events. The current widget set contains:

- `LayerLegendPanel`: a panel that shows the title/legend of a layer and some layer controls
- `MapLegendPanel`: a container of `LayerLegendPanel` instance, one for each map layer

We also have the following map controllers/listeners:

- `FeatureClickedListener`: a listeners that sends an `FeatureClickedEvent` when the user clicks on a feature
- `MouseListener`: a listeners that that sends an `FeatureMouseEvent` when the user hovers over a feature

---

# Chapter 2. Configuration

## 1. Dependencies

Make sure you include the plug-in in your project. If you are using Maven, add the following dependency to your pom:

### Example 2.1. Plug-in dependency

```
<dependency>
  <groupId>org.geomajas.plugin</groupId>
  <artifactId>geomajas-client-gwt2-plugin-corewidget</artifactId>
  <version>2.1.1</version>
</dependency>
```

---

# Chapter 3. How-to

## 1. Replacing the default style through GWT

To replace the default style, define another implementation for the client resource bundle factory through deferred binding:

```
<replace-with class="org.geomajas.gwt2.plugin.corewidget.client.resource.MyCoreWidgetClientBundleFactory"
  <when-type-is class="org.geomajas.gwt2.plugin.corewidget.client.resource.CoreWidgetClientBundleFactory"
</replace-with>
```

MyCoreWidgetClientBundleFactory should implement the same methods but use different css files and images.

## 2. Replacing the default style through css

To replace the default style through css, define the nostyle implementation for the client resource bundle factory through deferred binding:

```
<replace-with class="org.geomajas.gwt2.plugin.corewidget.client.resource.CoreWidgetClientBundleFactoryNoStyle"
  <when-type-is class="org.geomajas.gwt2.plugin.corewidget.client.resource.CoreWidgetClientBundleFactory"
</replace-with>
```

CoreWidgetClientBundleFactoryNoStyle is an implementation that provides empty css classes for all the css used. By importing a css file at the web page level, the web developer can now define his or her own style for each of these classes. The default css can be found at `org/geomajas/gwt2/plugin/corewidget/client/resource/CoreWidget.css`.

## 3. Replacing the complete view of one or more widgets

To replace the view completely, define another implementation for the view factory through deferred binding:

```
<replace-with class="org.geomajas.gwt2.plugin.corewidget.client.MyCoreWidgetViewFactory"
  <when-type-is class="org.geomajas.gwt2.plugin.corewidget.client.CoreWidgetViewFactory"
</replace-with>
```

The factory has create methods for each widget's view, just replace the default implementations by your own implementation classes.

---

# Chapter 4. Usage examples