IE Agency // Sitecore Module

Google Maps for Sitecore

DOCUMENTATION

prepared for Sitecore Shared Source
by Heiko Franz
on 10/11/2011
version 1.0

178 Albert Street
Windsor, VIC 3181

Tel 03 9001 1700
Fax 03 9529 8490
ie.com.au

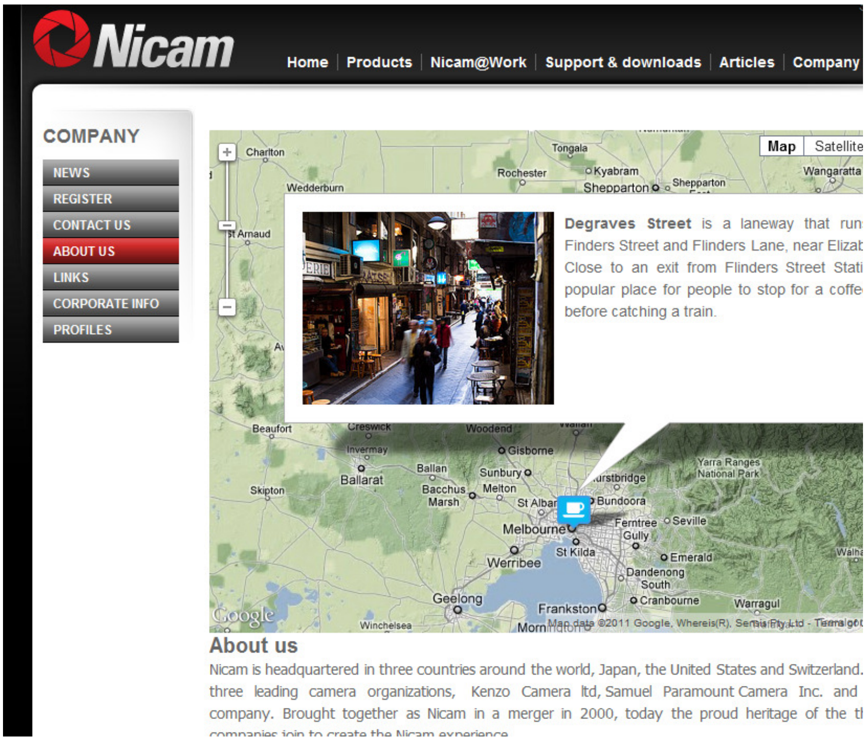
Table of Contents

1. Preface

This module allows to add custom maps to a Sitecore project to display office locations, tour stops or similar points of interest. It integrates the Google Maps API with Sitecore and currently supports the following:

- Add markers, lines and polygons
- Add info windows to markers and polygons
- Use custom icons for markers
- User friendly administration interface
- Fully customise the map (enable or disable any control, set maximum and minimum zoom level, ...)
- Flexible and extensible - use a custom Sublayout to match the requirement of your project without having to rebuild the module
- Rely on the current version 3 of the Google Maps JavaScript API
- Designed to work in Sitecore 6.5 - tested in Sitecore 6.2, 6.3 and 6.4

The module is a redevelopment of the Sitecore Maps module initially published by Timothy Ward in 2009.



Degraves Street is a laneway that run Finders Street and Flinders Lane, near Elizab Close to an exit from Flinders Street Stati popular place for people to stop for a coffe before catching a train.

About us

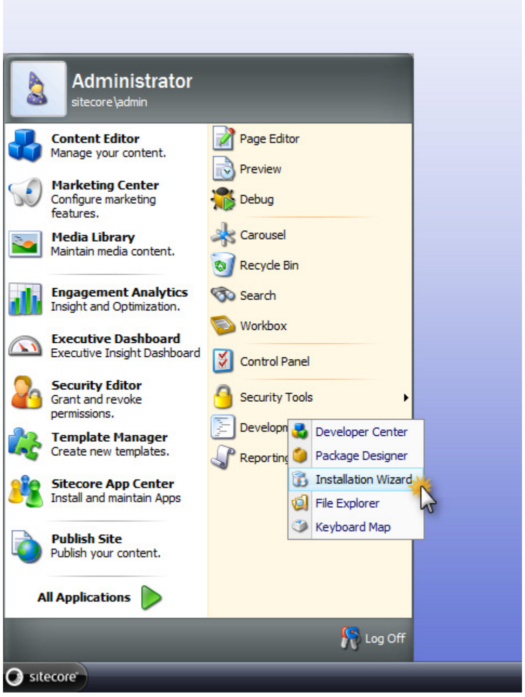
Nicam is headquartered in three countries around the world, Japan, the United States and Switzerland. three leading camera organizations, Kenzo Camera ltd, Samuel Paramount Camera Inc. and company. Brought together as Nicam in a merger in 2000, today the proud heritage of the tl companies join to create the Nicam experience

2. Introduction

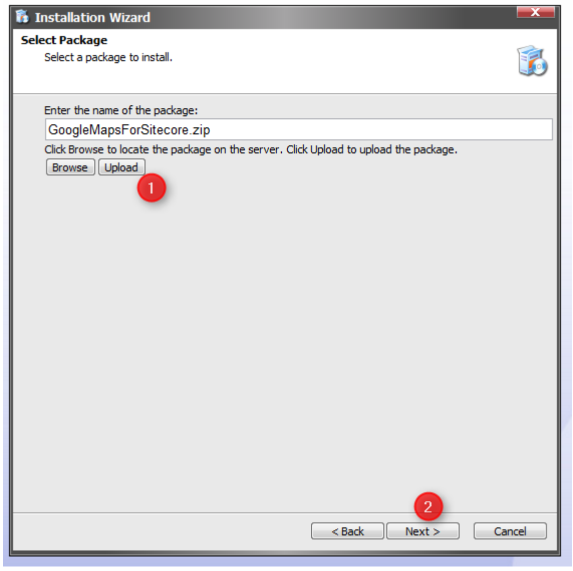
2.1. How to install

Always backup your Sitecore installation and database before installing a new module.

1. Download the module package and browse to the Desktop of your Sitecore installation.
2. Launch the Installation Wizard from the Development Tools:



3. Upload and install the package. Chose to overwrite existing items or files if you have used a previous version of the module.



Installation Wizard

Select Package

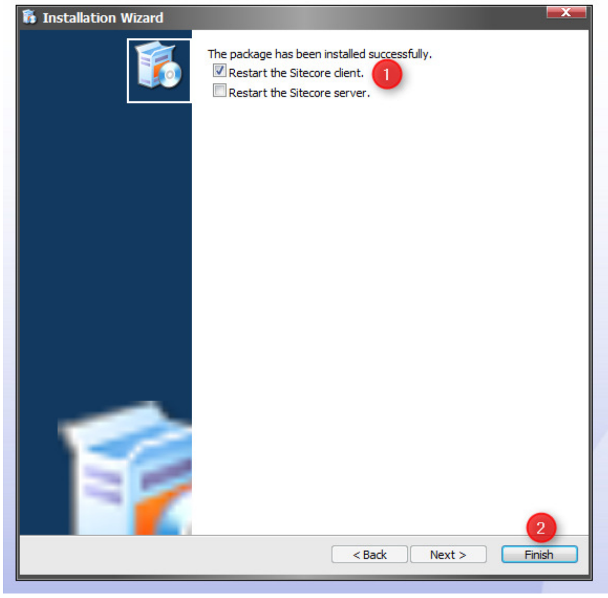
Select a package to install.

Enter the name of the package:

GoogleMapsForSitecore.zip

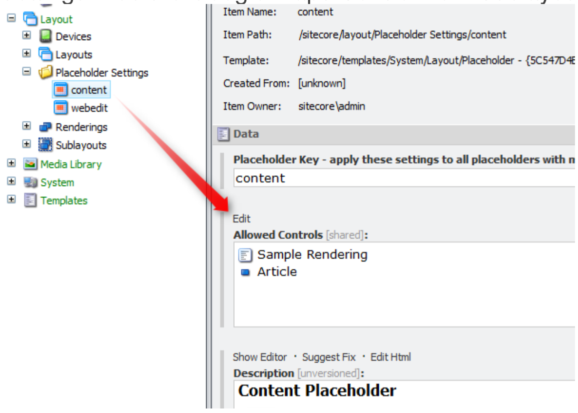
Click Browse to locate the package on the server. Click Upload to upload the package.

4. Restart the Sitecore client after the installation was successful.



The package has been installed successfully.
☑ Restart the Sitecore client.
□ Restart the Sitecore server.

5. Add the newly added Sublayout to the list of allowed controls for every placeholder that you want to hold a map. To do this, open the Content Editor and browse to /sitecore/Layout/Placeholder Settings. Add the Google Maps Control from Sublayouts / Modules:

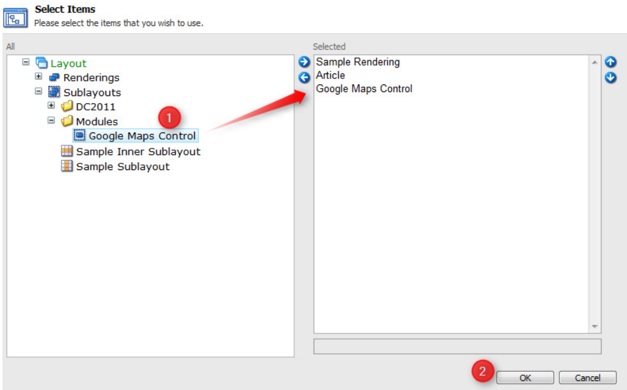


Item Name: content
Item Path: /sitecore/layout/Placeholder Settings/content
Template: /sitecore/templates/System/Layout/Placeholder - {5C547D4E
Created From: [unknown]
Item Owner: sitecore\admin

Placeholder Key - apply these settings to all placeholders with n
content

Allowed Controls [shared]:
Sample Rendering
Article

Description [unversioned]:
Content Placeholder



Select Items
Please select the items that you wish to use.

Selected:
Sample Rendering
Article
Google Maps Control

6. Publish the following items:
- Sublayouts\Modules\Google Maps Control
- Media Library\Google Maps Data (if you want to use the demo map)
- Content\Maps (if you want to use the demo map)

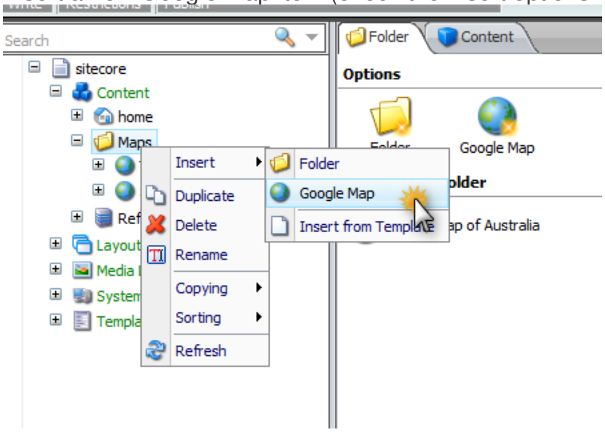
The module is now ready to use.

The following chapters explain how to configure a new map and how to add it to a page. Advanced topics like custom icons are outlined in later chapters.

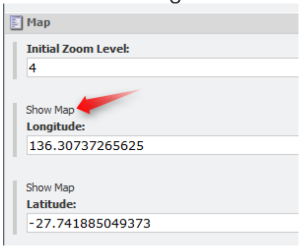
2.2. How to configure a map

Maps are configured using the content editor. By default, all maps are stored in \sitecore\Content\Maps (but you can store them anywhere in your project). The following explains how to add a new map:

1. Browse to \sitecore\Content\Maps
2. Insert a new Google Map item (check the insert options if this is not available)



3. Adjust the settings of the map. You can specify:
   - ○ The initial position and zoom level of the map. Click "Show Map" to find a location and zoom level using the editor



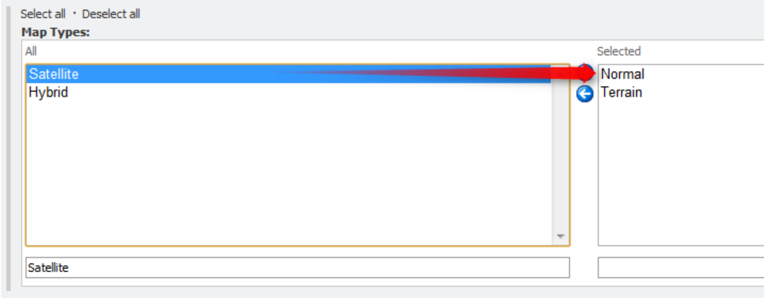
Map

Initial Zoom Level:
4

Show Map
Longitude:
136.30737265625

Show Map
Latitude:
-27.741885049373

   - ○ What map types to you want to use? The top one is the initial map type. The map defaults to a roadmap if no type is selected.



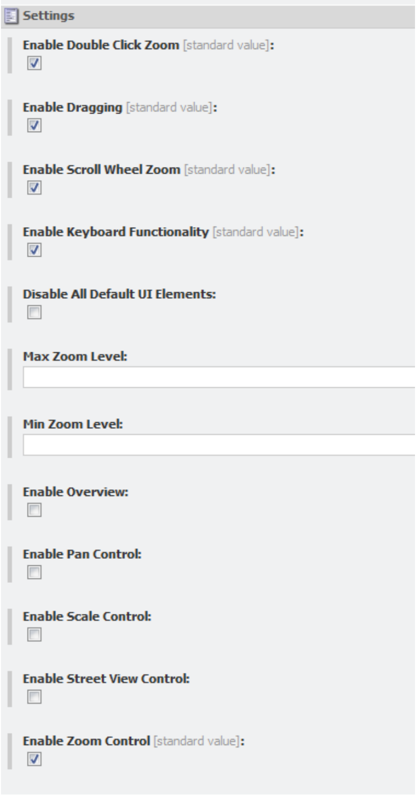
Select all · Deselect all
Map Types:

All	Selected
Satellite	Normal
Hybrid	Terrain

Satellite

   - ○ The size of the map on the page
   - ○ (optional) A custom cursor for dragging

- (optional) custom settings for the map



Settings

Enable Double Click Zoom [standard value]:
☑

Enable Dragging [standard value]:
☑

Enable Scroll Wheel Zoom [standard value]:
☑

Enable Keyboard Functionality [standard value]:
☑

Disable All Default UI Elements:
□

Max Zoom Level:

Min Zoom Level:

Enable Overview:
□

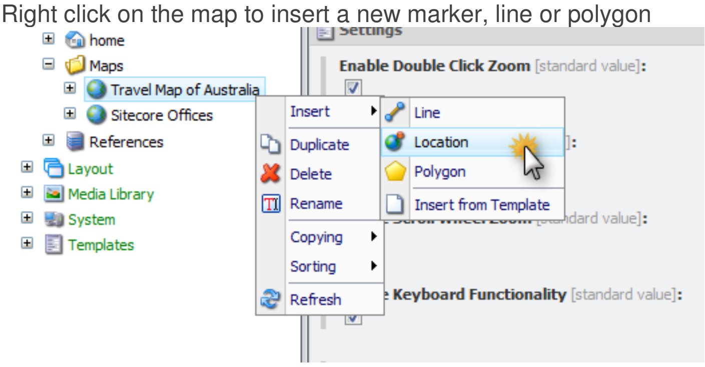
Enable Pan Control:
□

Enable Scale Control:
□

Enable Street View Control:
□

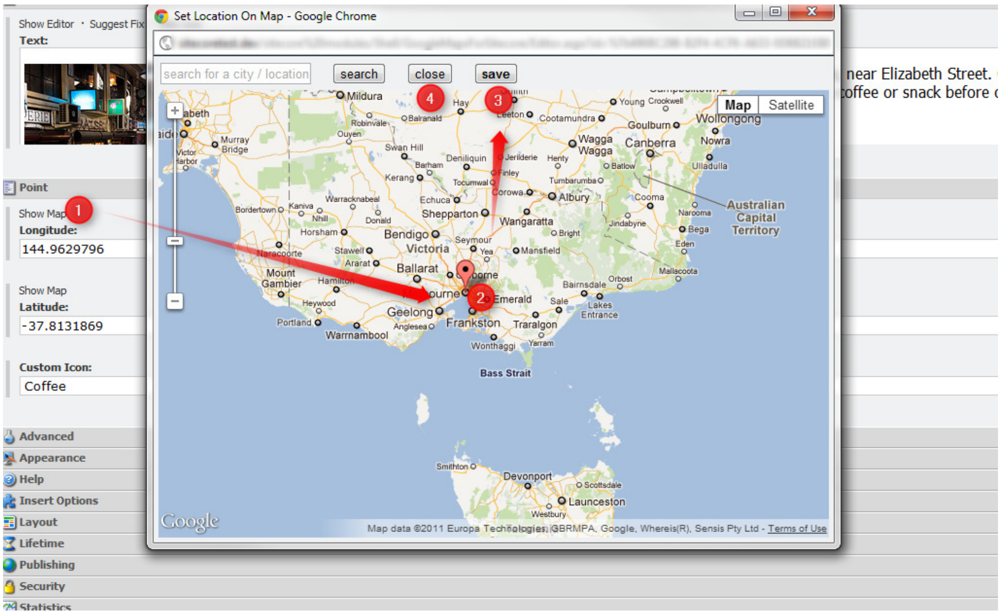
Enable Zoom Control [standard value]:
☑

4. Right click on the map to insert a new marker, line or polygon



5. (optional) Enter content for a info window for the marker (to appear in a box when you click on the marker) in the "Text" field. You can use any HTML, add images and links.

6. Click "Show Map" to specify the location of the marker on the map. Click on a location on the map, then click save and close.



Show Editor · Suggest Fix

Text:

near Elizabeth Street.
coffee or snack before c

Point

Show Map

Longitude:

144.9629796

Show Map

Latitude:

-37.8131869

Custom Icon:

Coffee

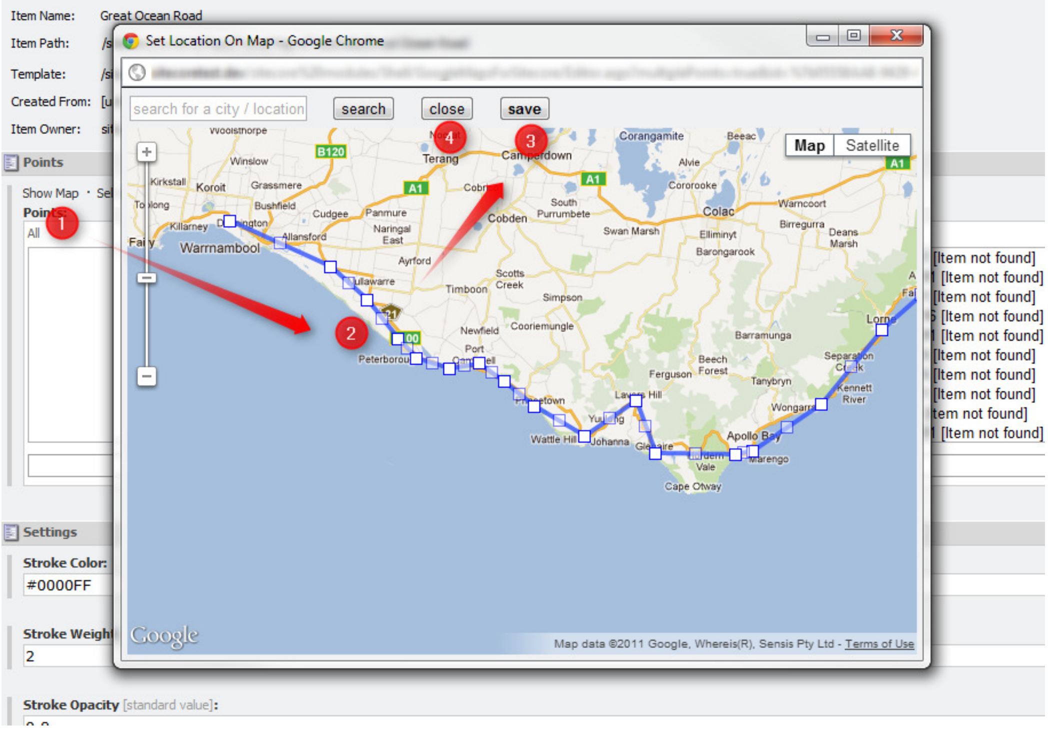
Advanced
Appearance
Help
Insert Options
Layout
Lifetime
Publishing
Security
Statistics

Set Location On Map - Google Chrome

7. You can search for an address or location using the search box in the top left.

8. You can pick a custom icon from the list if icons. The default red drop pin will be used if no icon is specified. You can define your own custom icons. See the chapter in the advanced topics for details.

9. Lines and polygons consist of multiple points. Click on "Show Map" to display an editor. You can add points to the map by clicking, move them around or remove one point by clicking on it. Click save and close to finish editing.



Item Name: Great Ocean Road

Item Path:

Template:

Created From:

Item Owner:

Points

Show Map · Se

Points:

All

[Item not found]
[Item not found]
[Item not found]
[Item not found]
[Item not found]
[Item not found]
[Item not found]
[Item not found]
tem not found]
[Item not found]

Settings

Stroke Color:

#0000FF

Stroke Weight

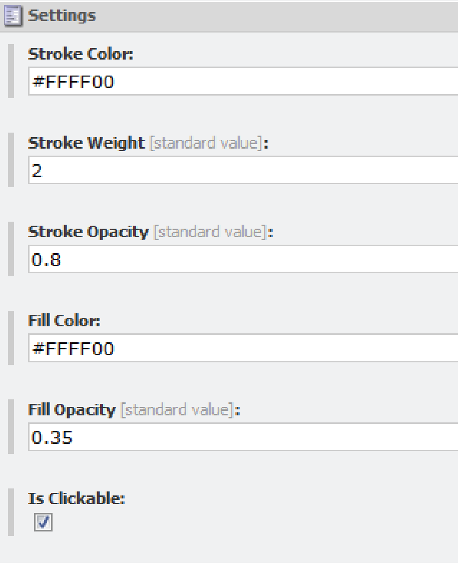
2

Stroke Opacity [standard value]:

Set Location On Map - Google Chrome

10. You can specify the stroke colour (as HTML/CSS colour code), the stroke weight (pixel value) and stroke opacity (between 0 and 1). Additionally, you can specify the fill colour and opacity for a

polygon and whether the polygon is clickable.



Settings

Stroke Color: #FFFF00

Stroke Weight [standard value]: 2

Stroke Opacity [standard value]: 0.8

Fill Color: #FFFF00

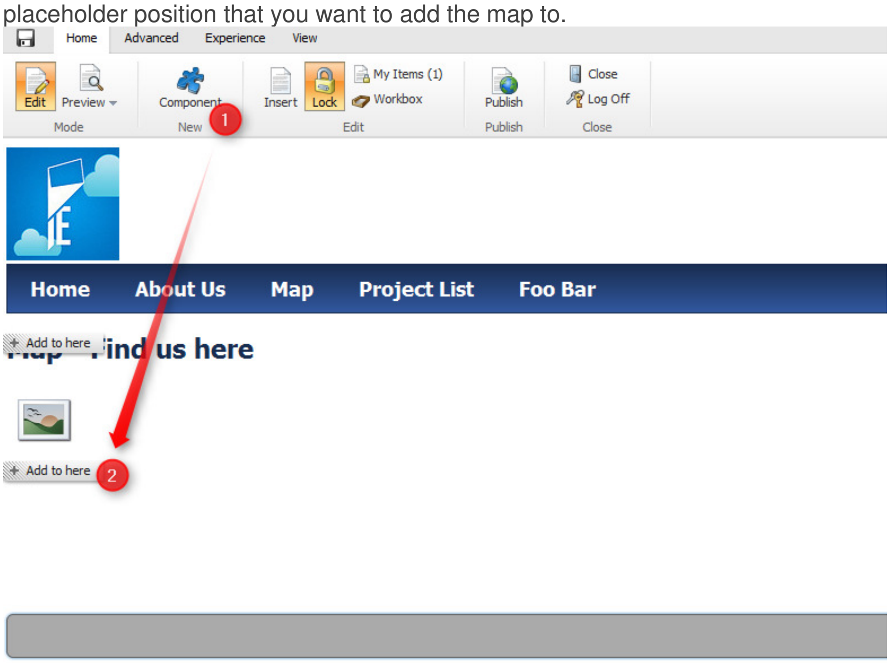
Fill Opacity [standard value]: 0.35

Is Clickable: ☑

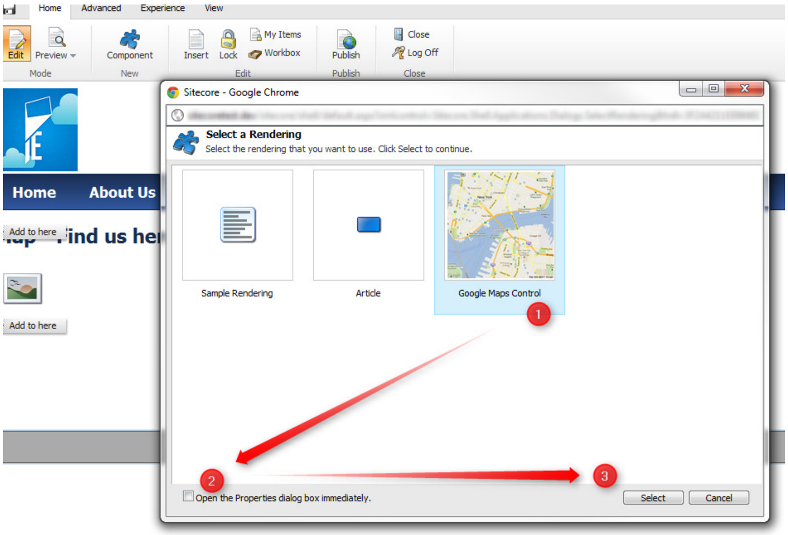
2.3. How to add a map using the Page Editor

Once a map is configured, it can be added to a page using the Page Editor or Content Editor. If you are using the Page Editor, browse to the page that you want to add the map to.

1. Click "Component" in the "Home" tab to add a new component. Click "Add to here" at the placeholder position that you want to add the map to.



2. Select "Google Maps Control" from the list of renderings. Check your placeholder settings if the Google Map control is not in the list. Tick the box "Open the Properties dialog box immediately" if you are using Sitecorein version 6.3 or earlier.

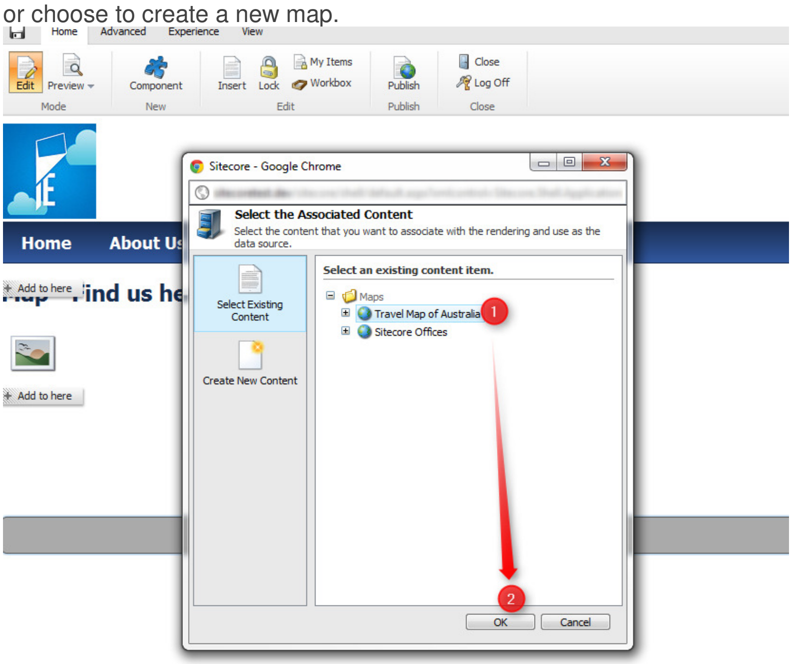


Select a Rendering

Select the rendering that you want to use. Click Select to continue.

□ Open the Properties dialog box immediately.

3. If you are using Sitecore 6.4 or newer, you are able to select a map to display in the following dialog or choose to create a new map.



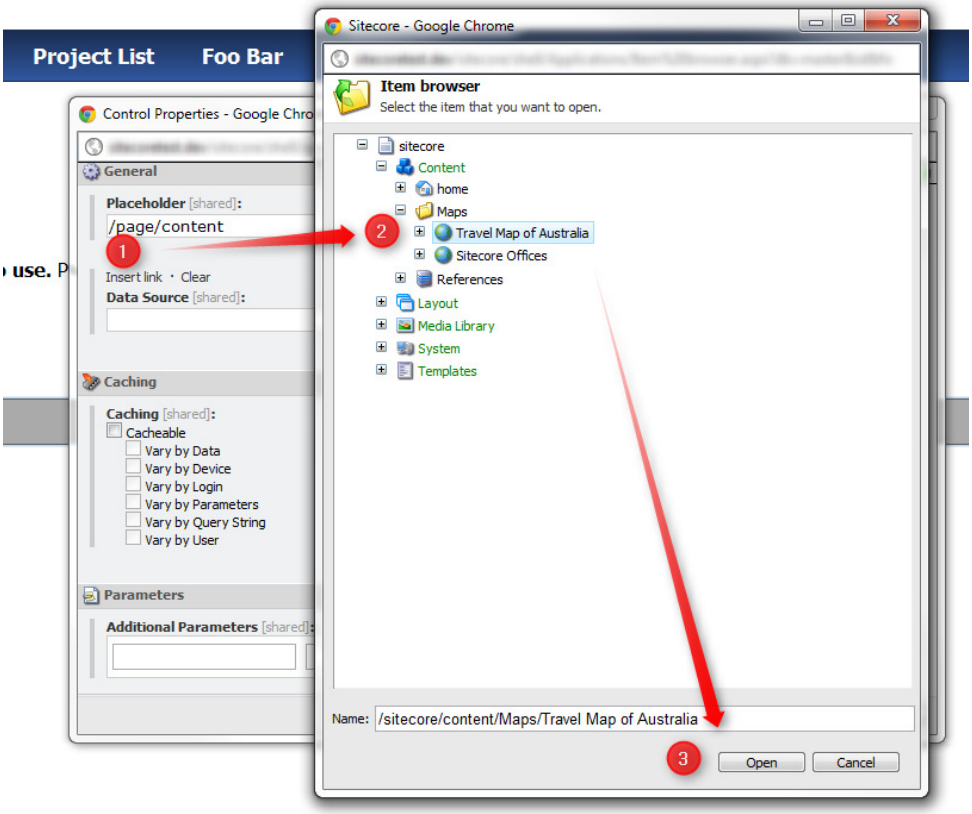
Select the Associated Content

Select the content that you want to associate with the rendering and use as the data source.

Select an existing content item.

4. For older versions of Sitecore, you have to manually set the "Data Source" in the Sublayout properties. Click on "Insert link" and select the map that you created earlier in

\sitecore\Content\Maps

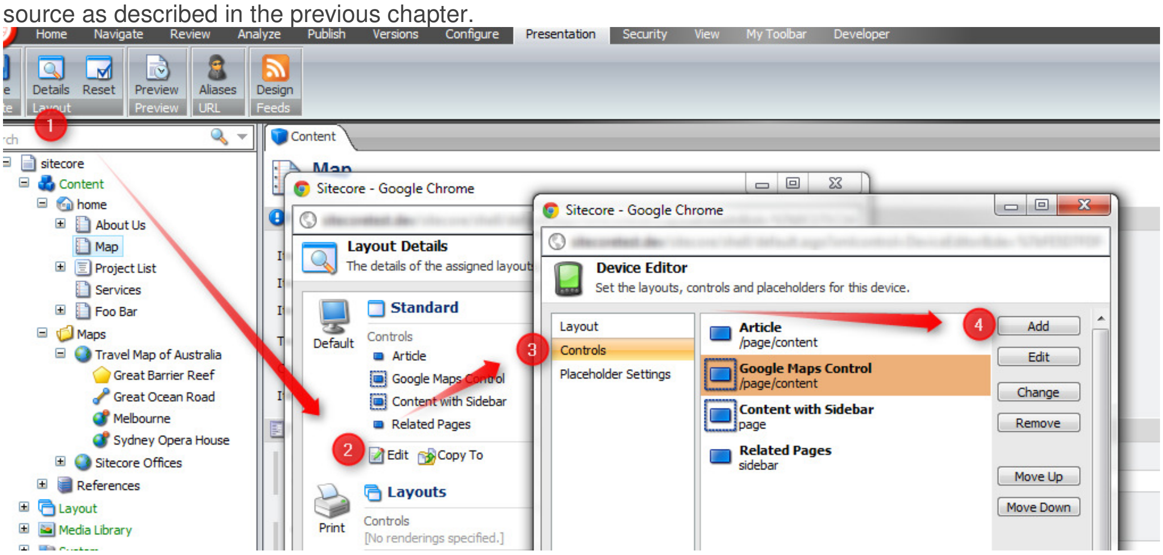


Placeholder [shared]: /page/content

Name: /sitecore/content/Maps/Travel Map of Australia

2.4. How to add a map using the Content Editor

To add a map using the Content Editor browse to the item that should display the map and click "Details" in the "Presentation" tab of the ribbon. Click "Edit" for the appropriate device, select "Controls" and "Add". Add the Sublayout from Sublayouts\Modules\Google Maps Control to the desired placeholder and set the data source as described in the previous chapter.



3. Advanced Topics

3.1. How to use custom icons

The Google Maps API supports custom icons for markers on a map. See http://www.powerhut.co.uk/googlemaps/custom_markers.php for details.

All custom icons are stored in \sitecore\System\Modules\Google Maps Settings\Custom Icons.

An icon consists of

- An image - PNG format
- A drop shadow image (optional) - PNG format
- Anchor - x and y position of where the icon is "pointing to" on the map
- Shadow Anchor - x and y position to anchor the shadow
- Clickable polygon - limits the area of the icon that is click-able, enter as a list of x and y positions

See the Google Maps API for more details and examples on custom icons.

3.2. Using multiple versions and languages

The module supports multiple versions and languages for maps just like any other item in Sitecore. Simply select the version and language from the "Versions" tab in the ribbon.

When translating a map to a new language, ensure that you transfer all module items into that language (including folders, settings, map types).

3.3. What is installed?

The module installed the following components:

- Files:
  - \bin\GoogleMapsForSitecore.dll
    the code for both backend and frontend
  - \sitecore modules\Web\GoogleMapsForSitecore\GoogleMap.ascx
    the usercontrol / sublayout rendering a map
  - \sitecore modules\Shell\GoogleMapsForSitecore\Editor.aspx, and some .png files
    the editor used in the Sitecore console to edit markers or other elements on the map
  - \App_Config\Include\GoogleMapsForSitecoreCommands.config
    configuration file for the custom commands on the map
- Core Database - 2 new field types to store a single point on the map as latitude and longitude as well as a multilist field that stores multiple points for a line and polygon.
- Master Database

- Templates in Templates\Google Maps
- Settings in System\Modules\Google Maps Settings
- 2 images in Media Library\Google Maps Data
- 1 new Sublayout in Layout\Sublayouts\Modules\Google Maps Control
- demo map with markers etc in Content\Maps

3.4. How to change the rendering of a map

It is possible to create a custom front end for a map without having to rebuild the module. Simply create a Sublayout / user control that inherits from GoogleMapsForSitecore.Layouts.GoogleMap. This class has a public property CurrentMap of type GMap that contains all data needed to display the map. To display the map using JavaScript, simply call GoogleMapsForSitecore.Utilities.JSON(CurrentMap) to turn the map into a JSON object.

3.5. How to build

This project needs to be built against a Sitecore instance. Update the <SitecorePath> property in the project files GoogleMapsForSitecore\GoogleMapsForSitecore.csproj to point at the website folder of an existing Sitecore instance (tested with Sitecore 6.5).

The following files need to be copied across to the Sitecore installation to test the build:

- \bin\GoogleMapsForSitecore.dll
- \sitecore modules\Web\GoogleMapsForSitecore\GoogleMap.ascx
- \sitecore modules\Shell\GoogleMapsForSitecore\Editor.aspx, all .png files
- \App_Config\Include\GoogleMapsForSitecoreCommands.config

The project contains post build action that copies the files across. View the bottom of the GoogleMapsForSitecore.csproj for details and to disable the copy target if not needed.

Take the following steps to make the project part of an existing Visual Studio project with Sitecore:

- install the module as described in this document.
- put the GoogleMapsForSitecore.dll in a dependencies folder and reference it in your project.
- use the GoogleMap sublayout that comes with this module or create your own one by creating a usercontrol that inherits from GoogleMapsForSitecore.Layouts.GoogleMap as described above.

4. Known Issues / Limitations

It is currently not possible to change the content or settings of a map within the Page Editor, this feature might be added in later versions.

The Google Maps API usage limitations apply. Refer to http://code.google.com/apis/maps/faq.html#usagelimits

Item Not Found - When working with lines or polygons, the field type shows [Item Not Found] in the list next to all points. This is a known bug and does not affect the module in any way.

The heavy usage of Javascript on the map can slow down the Page Editor. A later version of this module can introduce a setting that disables the map while the page is open for editing in Page Editor mode.

The module supports maps in different languages (by creating map items in different languages) but there are currently no translations available for the Sitecore admin console of the module.

5. References / Remarks

Thanks to Timothy Ward at Sitecore Australia for building the initial version of this Google Maps module on Sitecore shared source.

Also thanks to the .NET team at IE Media for beta testing and providing feedback.

The following resources were used during development:

- Google Maps Javascript API V3 Reference
  http://code.google.com/apis/maps/documentation/javascript/reference.html
- Google Maps API V3 Enable Editing Polylines
  http://snipplr.com/view/38270/google-maps-api-v3--enableediting-polylines/
- Using the DataSource Filed with Sitecore Sublayouts - Mark Ursino
  http://firebreaksice.com/using-the-datasource-field-with-sitecore-sublayouts/