

Elixir Map Designer Manual

Release 7.7



Elixir Technology Pte Ltd

Elixir Map Designer Manual: Release 7.7

Elixir Technology Pte Ltd

Published 2009

Copyright © 2009 Elixir Technology Pte Ltd

All rights reserved.

Solaris, Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. Microsoft and Windows are trademarks of Microsoft Corporation. All other trademarks are registered under their respective owners.

Table of Contents

1. Introduction	1
Overview	1
Getting Started	1
2. Map Creation	2
Step 1 : Store the Shapefiles in the Repository	2
Step 2 : Create Map file from .shp file	2
Step 3 : Define Primary Key	4
Step 4 : Customise Base Map Format	4
Step 5 : Add Map Layer(s) to the Base Map	6
Step 6 : Final Format to Show a Complete Map	7
3. Data Integration	10
Converted to .ds file	10
4. Designer Features	11
Types of Colouring Algorithm	11
Array	11
Gradient 2	13
Gradient 3	15
Image	16
Lookup	17
Legend	18
Map Layers	20
Primary key	22

List of Figures

2.1. Store Shapefiles in the Repository	2
2.2. Define .shp file as .map file	3
2.3. Display default .map file	3
2.4. Select Primary Key	4
2.5. Select Appropriate Colouring Algorithm	5
2.6. Format Title	5
2.7. Format Legend	6
2.8. Format Map Layer	7
2.9. Enable Map Layer	7
2.10. Map After Show Data	8
2.11. Effect of Data on Colouring	9
3.1. Convert .dbf file to .ds file	10
4.1. Colouring Algorithm - Array (with few regions)	12
4.2. Colouring Algorithm - Array (with numerous regions)	13
4.3. Colouring Algorithm - Gradient 2	14
4.4. Colouring Algorithm - Gradient 3	16
4.5. Colouring Algorithm - Image	17
4.6. Colouring Algorithm - Lookup	18
4.7. A Sample of Continuous Legend	19
4.8. A Sample of Continuous Legend without Start At Zero Option	20
4.9. Example of Base Map without Map Layer	21
4.10. Example of a Base Map with Map Layers	22

Chapter 1

Introduction

Overview

Elixir Repertoire Map Designer provides a graphical framework that adds a new dimension to data visualisation. Maps can be created that represent geographic areas, such as countries or regions, shops within a shopping mall or even the different departments in your office. The areas of the map are sensitive to mouse clicks, so you can use them to navigate through or filter relevant data. The areas can also be connected to data values, highlighting which sales regions performed best last quarter or which shops have the highest footfall.

Once a map has been designed, it is available in the Elixir Repository as a reusable component. This helps you ensure a consistent style and reduces maintenance across all of the analysis services that you deploy.

Getting Started

Before you can start creating maps, you need map data for the location you want to see. This data consists of two files, a shapefile (.shp) and a database file (.dbf). The pair of files should have the same name, for example Singapore.shp and Singapore.dbf. The shapefile format was defined by ESRI.

The shapefile holds both the shapes and coordinates of the map, whereas the database file holds the meaning of those shapes and points. Both are essential for rendering the map. As the file format for shapefiles has been published, and is a de-facto standard, you can create your own shapefiles if you can't find one that exactly matches your requirements.

Tip

Useful links:

- Download of shapefiles [<http://www.cdc.gov/epiinfo/asia.htm>].
- Explanation of common data column [<http://www.cdc.gov/epiinfo/documents/shapes.txt>] in the .dbf file.

Important

While there are a number of shapefiles available on the Internet, you need to verify that the license allows you to incorporate them in your applications.

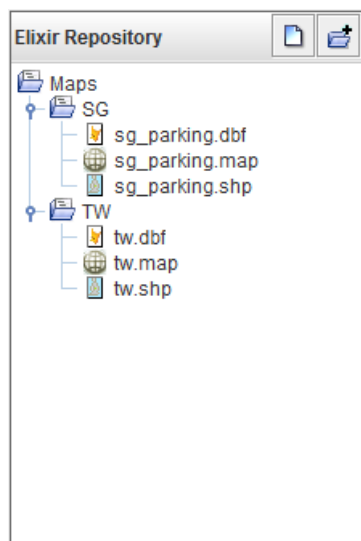
Chapter 2

Map Creation

Step 1 : Store the Shapefiles in the Repository

1. Save the .shp and .dbf files in a repository location where all users of the map will have read access to them.

Figure 2.1. Store Shapefiles in the Repository



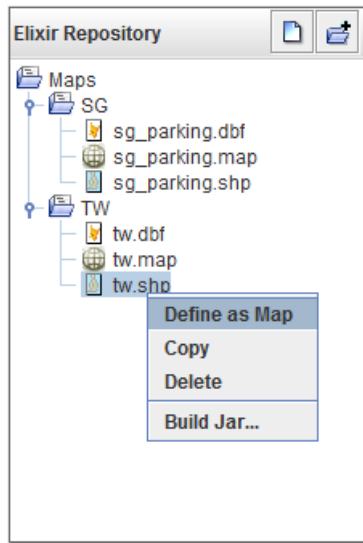
Note

- In order for the map to render successfully,
 - The shapefiles must be kept together in the same folder.
 - The naming of .shp and .dbf files must be the same.

Step 2 : Create Map file from .shp file

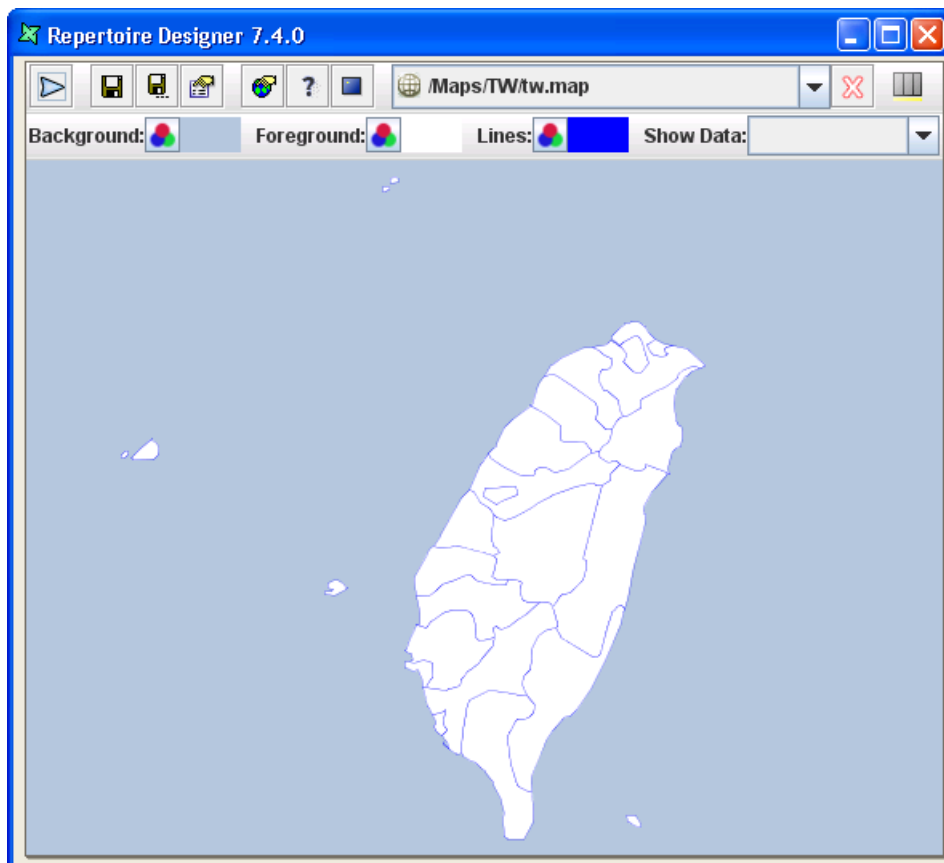
1. Select the .shp file that will form the background of your map, right-click and select *Define As Map*.

Figure 2.2. Define .shp file as .map file



2. A default .map file will be created and displayed in the Elixir Workspace as shown in Figure 2.3, "Display default .map file".

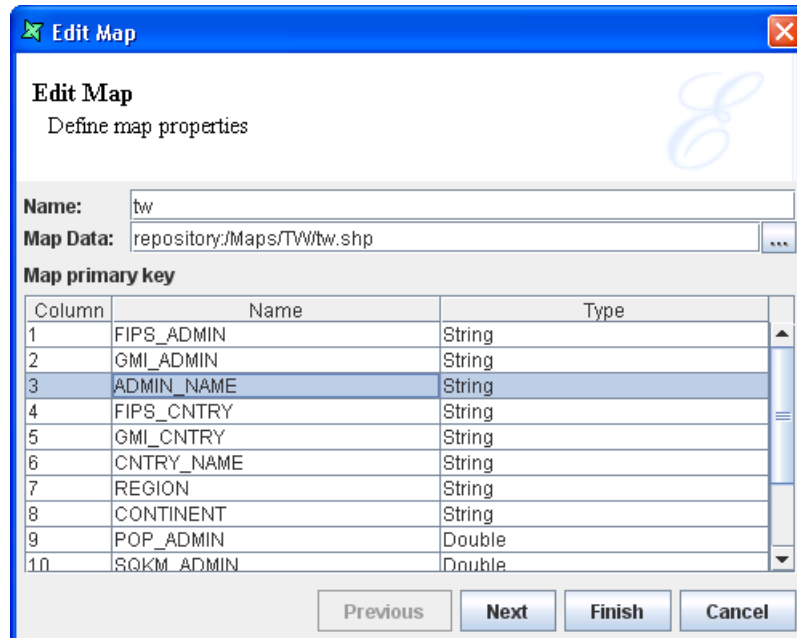
Figure 2.3. Display default .map file



Step 3 : Define Primary Key

1. Select the .map file and right-click to select Open Properties
2. Select an appropriate Primary key and click Next.

Figure 2.4. Select Primary Key



Tip

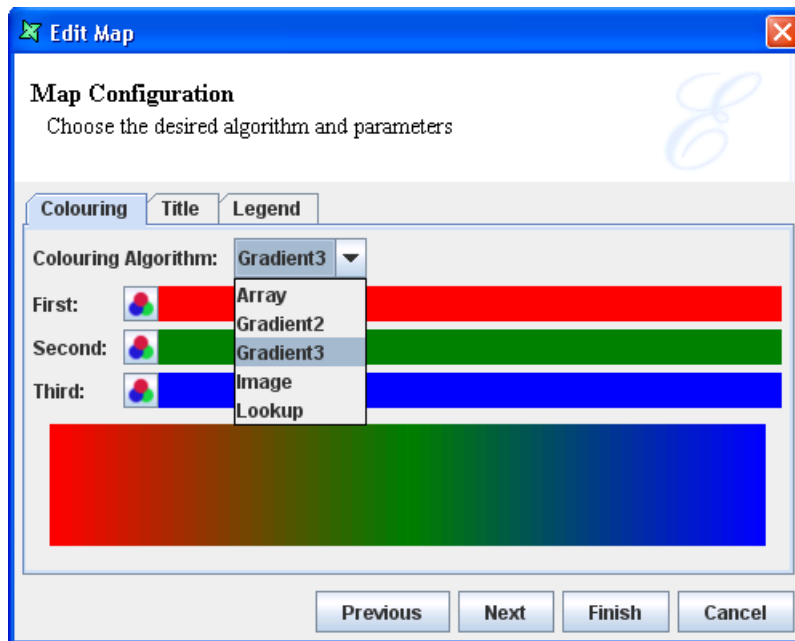
Why is *ADMIN_NAME* a more appropriate primary key?

If the base map has many regions, it may be a better choice to show as a hovering legend when the cursor is placed over each region.

Step 4 : Customise Base Map Format

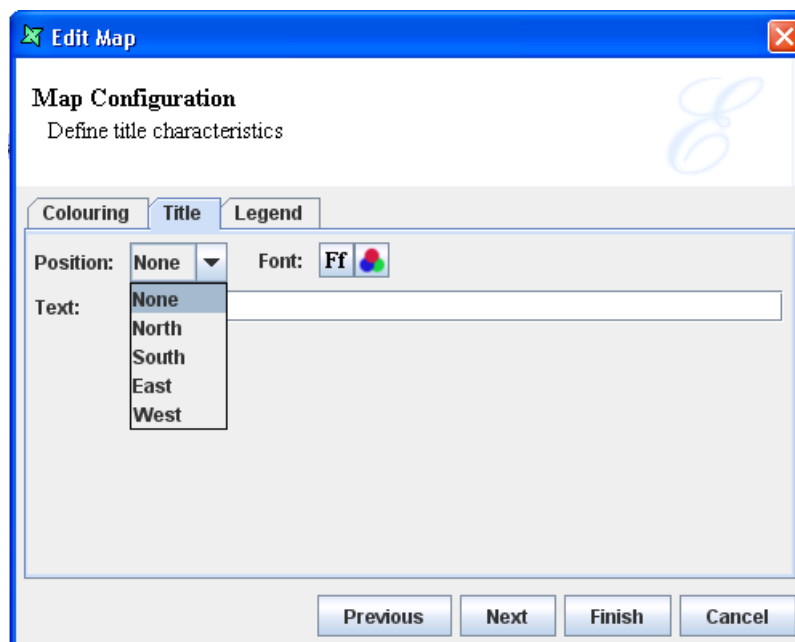
1. You will see the *Map Configuration* page is where user can customise the appearance of the map.
2. Under the *Colouring* tab, select the type of colouring algorithm desired for the respective regions of the map.

Figure 2.5. Select Appropriate Colouring Algorithm

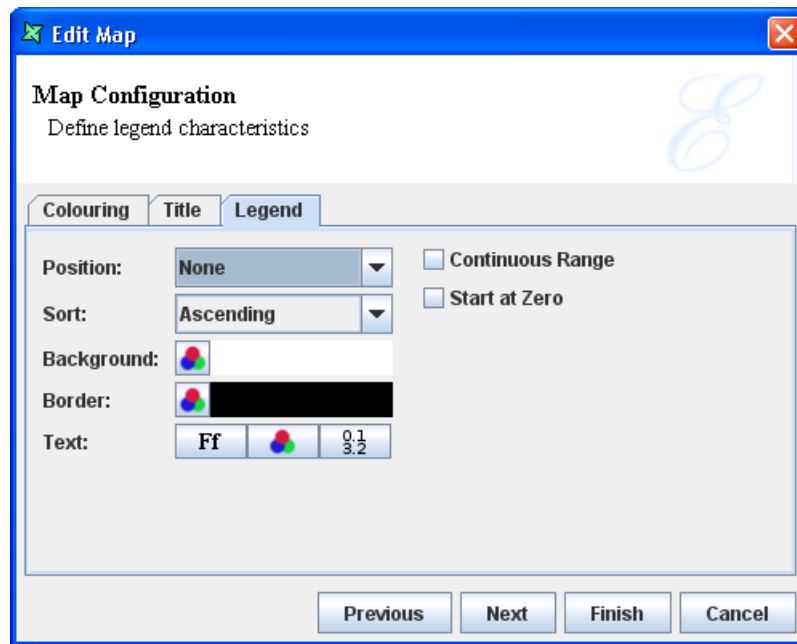


3. Under the `Title` Tab,
 - Text option : (Optional) Enter a title that will appear on your map.
 - Font option : (Optional) Select the size and colour of the title.
 - Position option : (Optional) Place the title at the top (north), bottom (south), east (right) or west (left) of the map.

Figure 2.6. Format Title



4. Under the `Legend` Tab, select the characteristics of the map legend.

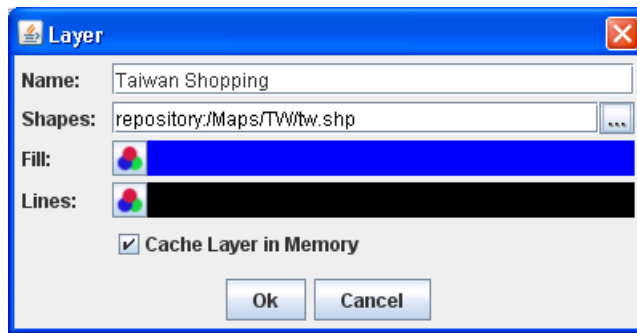
Figure 2.7. Format Legend**Tip**

- If you select None, user still be able to see a tooltip based on the map primary key when hovering the mouse cursor over a particular region.
- If there is handful of coloured regions on the map, user can consider showing colours derived from numeric data, you should consider by enabling the Continuous option (refer to Legend)

Step 5 : Add Map Layer(s) to the Base Map

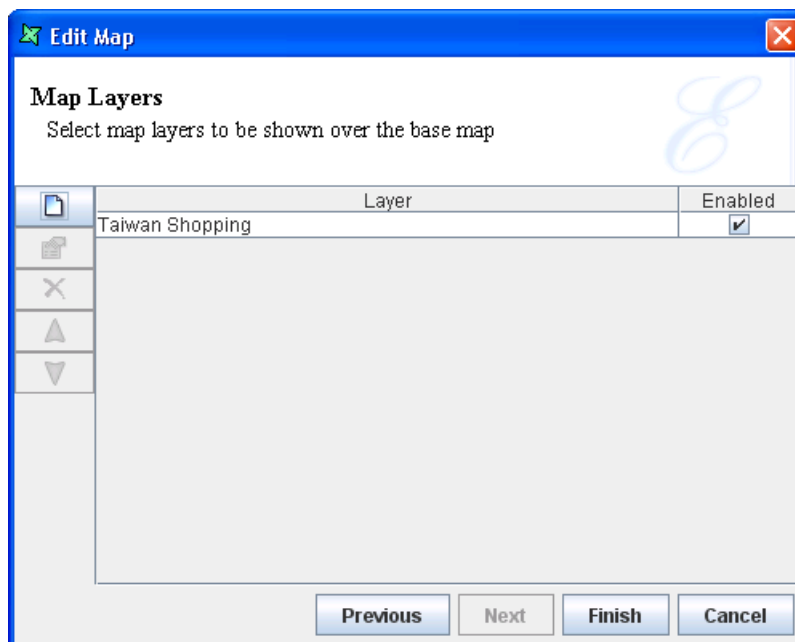
If you have Map Layers that you wish to add on to the base map, they can be configured here. If not, click on Finish button to complete the creation process.

1. Click the Add icon and enter the details as follows:
 - Name option : Enter a meaningful name for the map layer.
 - Shapes option : (As an example shown in Figure 2.8, "Format Map Layer") Enter the URL of the shapefile, for example, *repository:/MyMaps/Taiwan/tw_shoppingmalls.shp* .
 - Fill option : (Optional) Select the colour to appear in the area of the map layer.
 - Line option : (Optional) Select the colour for the outline of map layer, which is especially useful when representing railroad or highway routes which are usually represented by lines.

Figure 2.8. Format Map Layer**Tip**

User can select the layer map .shp file and select Copy to store the location path to the clipboard in order to paste it under the Shapes option.

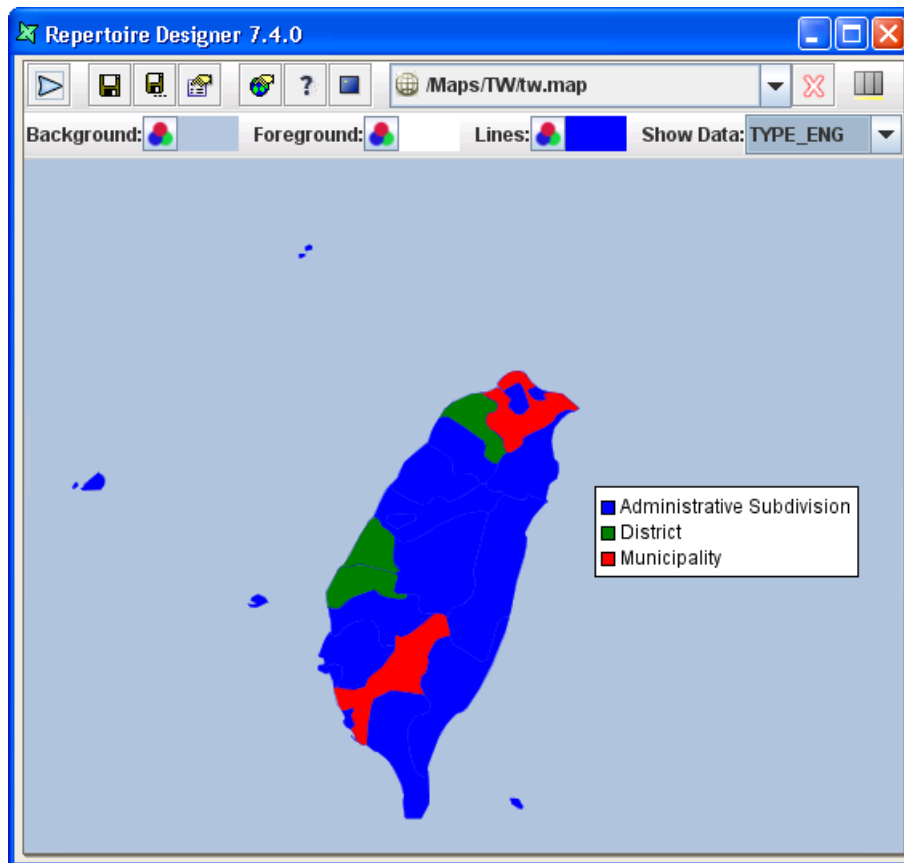
- When a map layer is added, check the check box to enable the map layer to appear as an overlay on the base map, as shown in Figure 2.9, “Enable Map Layer”. User can click on the Edit icon to modify the earlier configurations if necessary.

Figure 2.9. Enable Map Layer

Step 6 : Final Format to Show a Complete Map

If the base map has “fill-regions” data and a particular Colouring Algorithm has already been setup, user can now preview the map with data.

- Drag a datasource (the base map .ds file is a good place to start) to anywhere within the Map view and drop it. This is to populate Show Data : with the data from the .ds file.
- Select the field name in the drop down list to activate the Colouring Algorithm type chosen. The result will be similar to Figure 2.10, “Map After Show Data”.

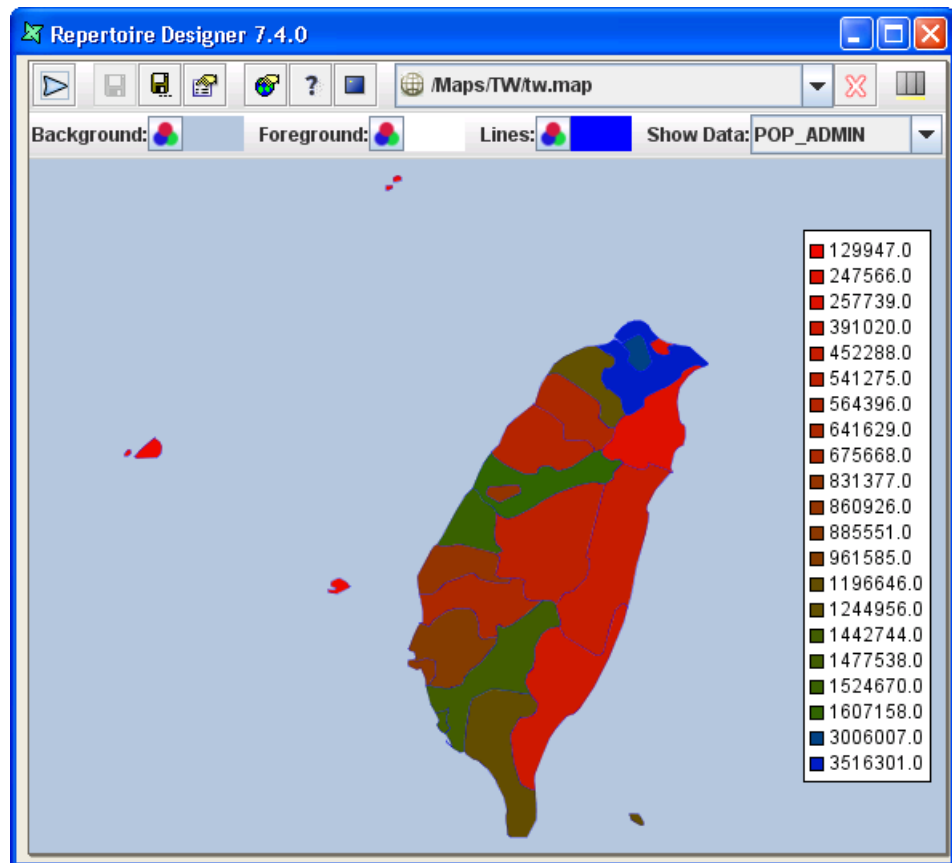
Figure 2.10. Map After Show Data**Tip**

If a “text” legend is required, user can use *GMI_ADMIN* as the next best option to describe all regions on the map.

3. When the mouse cursor hovers over any one region, the region's name will appear, as seen in Figure 2.10, “Map After Show Data”. User can select any field name to experiment how the map will appear with each selection.

Tip

In this example, an interesting field name, *TYPE_ENG*, shows a group of regions under a specific category. Thus, when selecting this field name, user can see that a number of regions are classified under the same colouring tone instead. In this case, Legend is best used to display in order to identify all the categories for each region.

Figure 2.11. Effect of Data on Colouring**Note**

GMI_ADMIN and *TYPE_ENG* are common field names found in many .dbf files. There are also some .dbf files which do not have such field names or have the same data with different field names.

4. On the top of Elixir Workspace, there are optional formatting tools to help further to enhance the image of your map.
 - Background option : Select the colour to appear outside the base map area.
 - Foreground option : Select the colour to appear inside the base map area.
 - Lines option : Select the colour for the outline of the regions

Tip

- The foreground colour is useful for specifying regions which do not have "fill-regions" data in the .ds file. It helps to contrast the rest of the coloured regions.
- It is also useful for a base map whose purpose is to show map layers clearly. Thus, the foreground colour should be one that can bring out the clarity of the points on the base map. (Refer to the section called "Map Layers").

Chapter 3

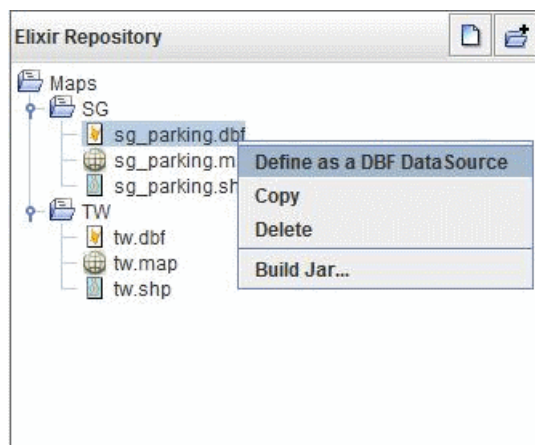
Data Integration

Convert .dbf file into .ds file

To show colours on a map, user need to choose a set of data and a colouring algorithm to describe how to convert the data into colours for display. The simplest source of data is the .dbf file that accompanies the .shp file. This contains the name of each region on the map, and often contains useful information such as area which can be used as sample data for testing. In order to use the .dbf, .dbf file will be to be converted to a .ds file.

1. Select a .dbf file and right-click to select `Define as a DBF DataSource`, as seen in Figure 3.1, “Convert .dbf file to .ds file”.

Figure 3.1. Convert .dbf file to .ds file



2. A default .ds file will be created
3. Click on the `Load Data` button to view data.

Note

Some shape files contain layer information, for example roads or hospitals. The .dbf files that accompany these shape files may not contain any useful region information for colouring. Using the Elixir data source tools, it is easy to review the contents of the .dbf files.

Chapter 4

Designer Features

Colouring Algorithm

This algorithm decides the colour scheme that will be used to fill each region of the base map.

Array

The Array algorithm allows user to specify a number of colours and use each in turn for a data item. This works best when the number of colours choosen is greater than or equal to the number of unique data values. Else, there will be some repetition of the colours.

For the field name selected in `Show Data`, the first region will take the first colour and so on until the 6th colour is utilised by the 6th region. The 7th region will have the same colour as the 1st region.

Note

- It is useful to apply in a map with only a few regions to fill the colour scheme as shown in Figure 4.1, “Colouring Algorithm - Array (with few regions)”.
- An example of inappropriate usage of this colouring algorithm is shown in Figure 4.2, “Colouring Algorithm - Array (with numerous regions)”. This colour scheme can still be used if specific colour type is important but not critical to differentiate each region.

Figure 4.1. Colouring Algorithm - Array (with few regions)

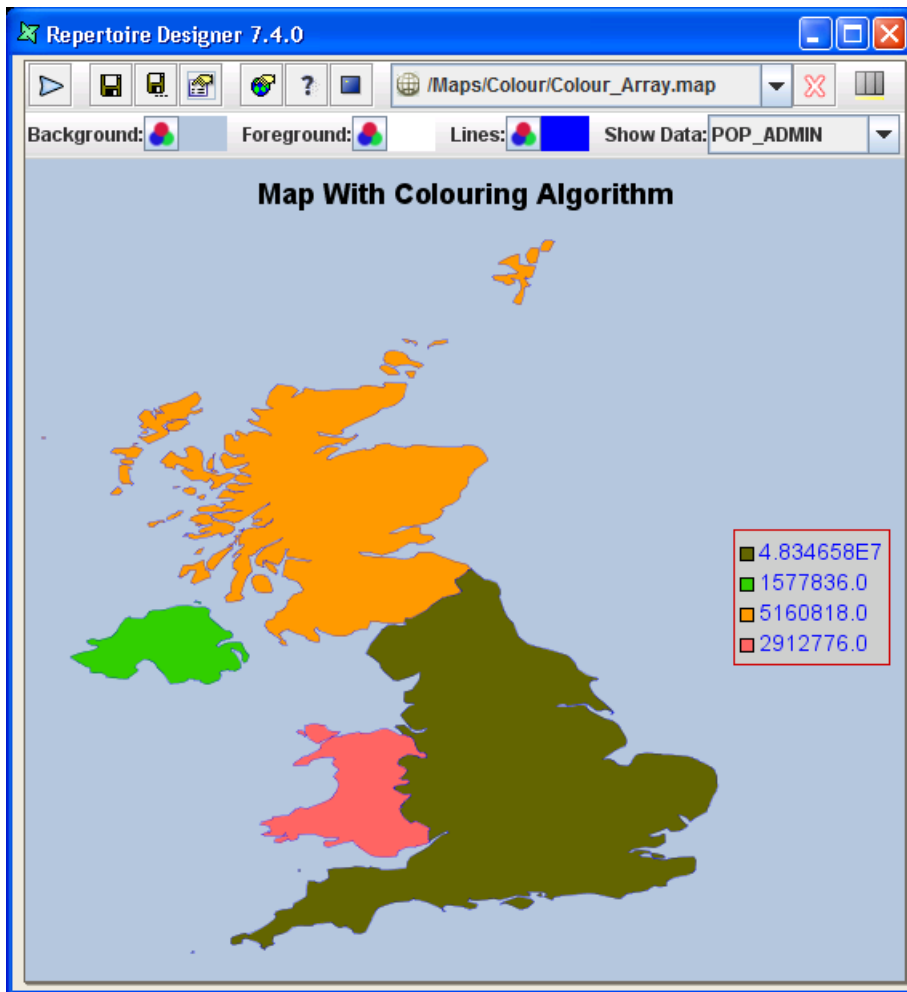
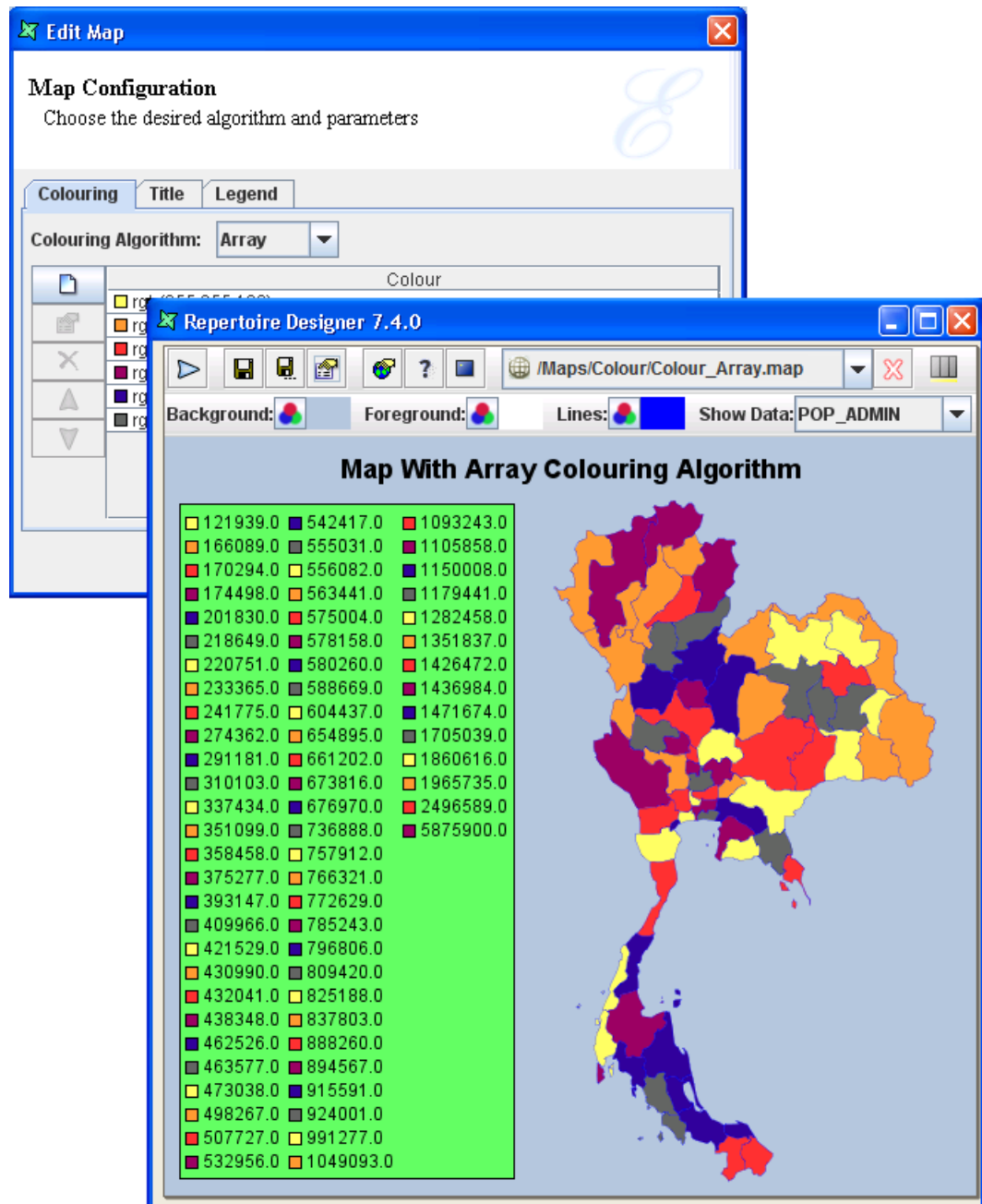


Figure 4.2. Colouring Algorithm - Array (with numerous regions)

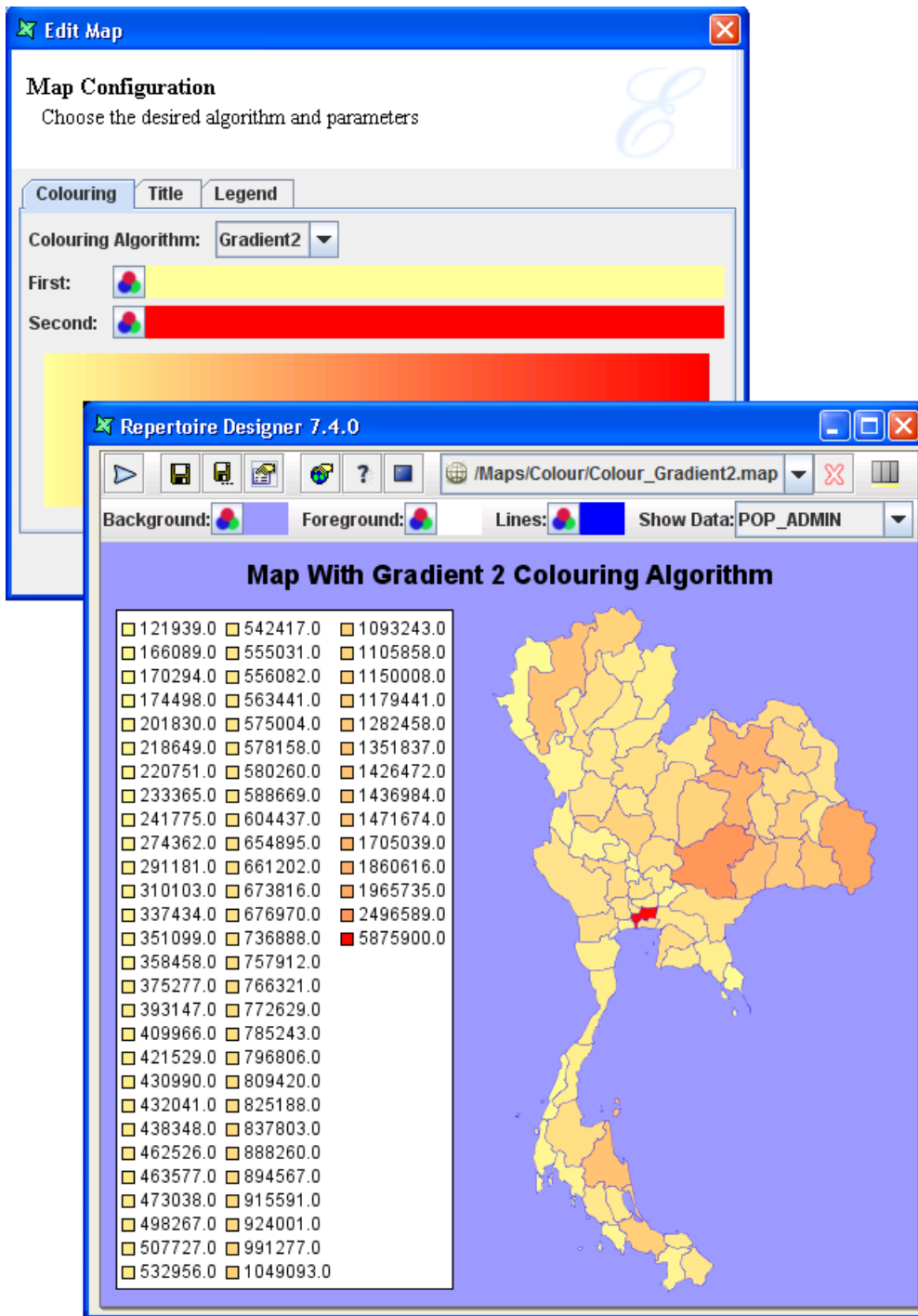


Gradient 2

A range of colours will be blended from the first colour specified to the second colour specified as shown in the Map Configuration page, Colouring tab.

For the field name you selected under the Show Data combo box, the first region will take on the first colour specified, subsequent regions will take on the increasing colour tone down the line until the last region will usually take on the second colour specified.

Figure 4.3. Colouring Algorithm - Gradient 2



Note

The legend displayed in Figure 4.3, “Colouring Algorithm - Gradient 2” is for the purpose of explaining the effect of the Gradient 2 on the colouring of the regions. There is no need to display such legend on map.

Tip

- This colour scheme is useful if user wishes to see an increasing colour tone in data to reflect the gradual effect of the regions.

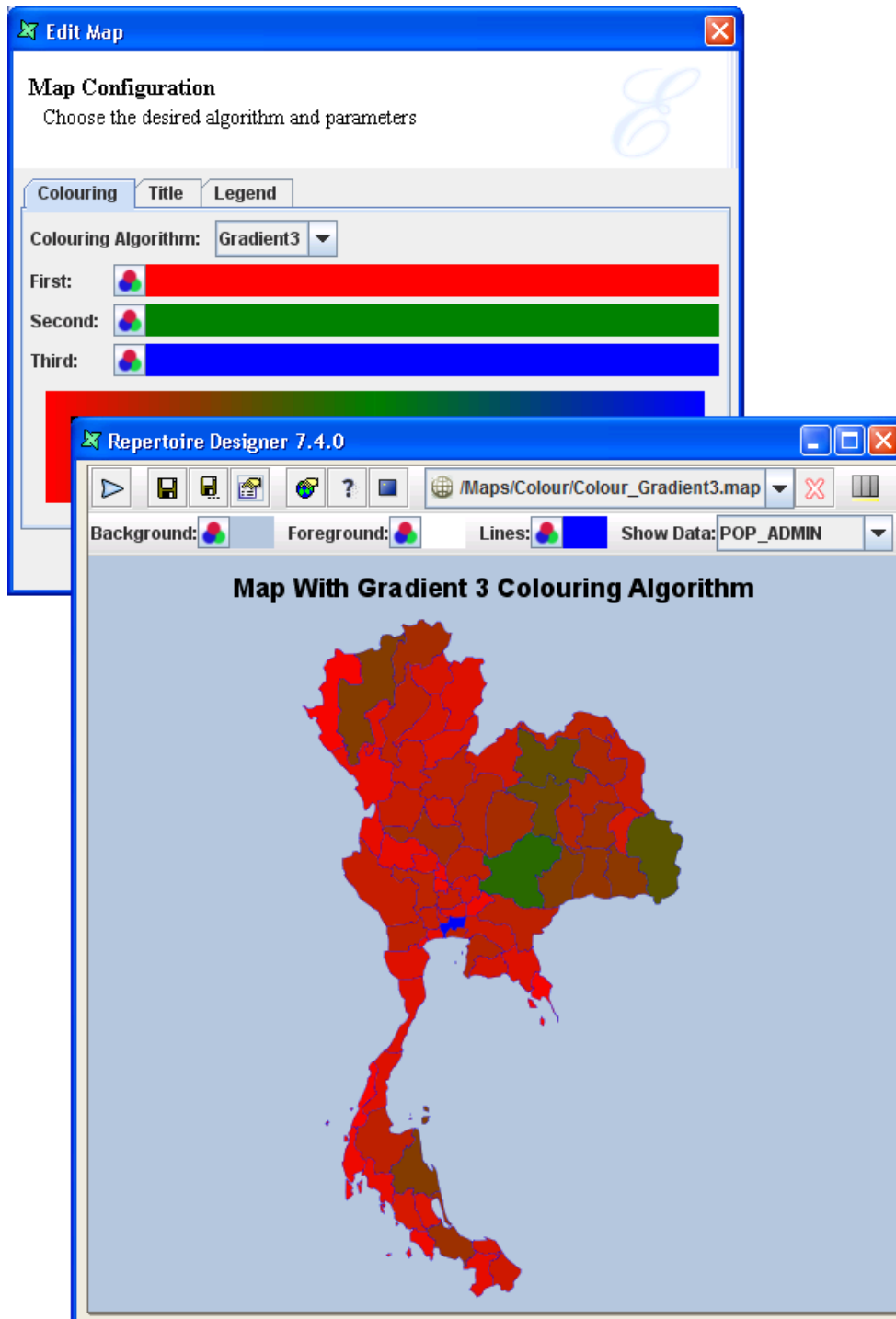
- An example like in Figure 4.3, “Colouring Algorithm - Gradient 2” *POP_ADMIN* is selected to show the regions of increasing population size compared to the size of the regions.
- There are many lighter tone coloured regions as compared to the darker ones, which indicates that 'small-sized populations' regions are very prominent in the base map.

Gradient 3

Gradient 3 is similar to Gradient 2, but the range of colour will be blended from first colour to second colour (median) to third colour instead.

This colour scheme is useful if user wishes to see a increasing colour tone for data reflect the gradual effects of the regions, especially to highlight regions that fall in the median range.

An example like in Figure 4.4, “Colouring Algorithm - Gradient 3”, *SQMI_ADMIN* is selected to show the regions of increasing area in square miles. Select a totally different colour tone as the median so as to differentiate from the first and third colour. At a glance, user will be able to see the green tone regions which indicates that they are about the same size each.

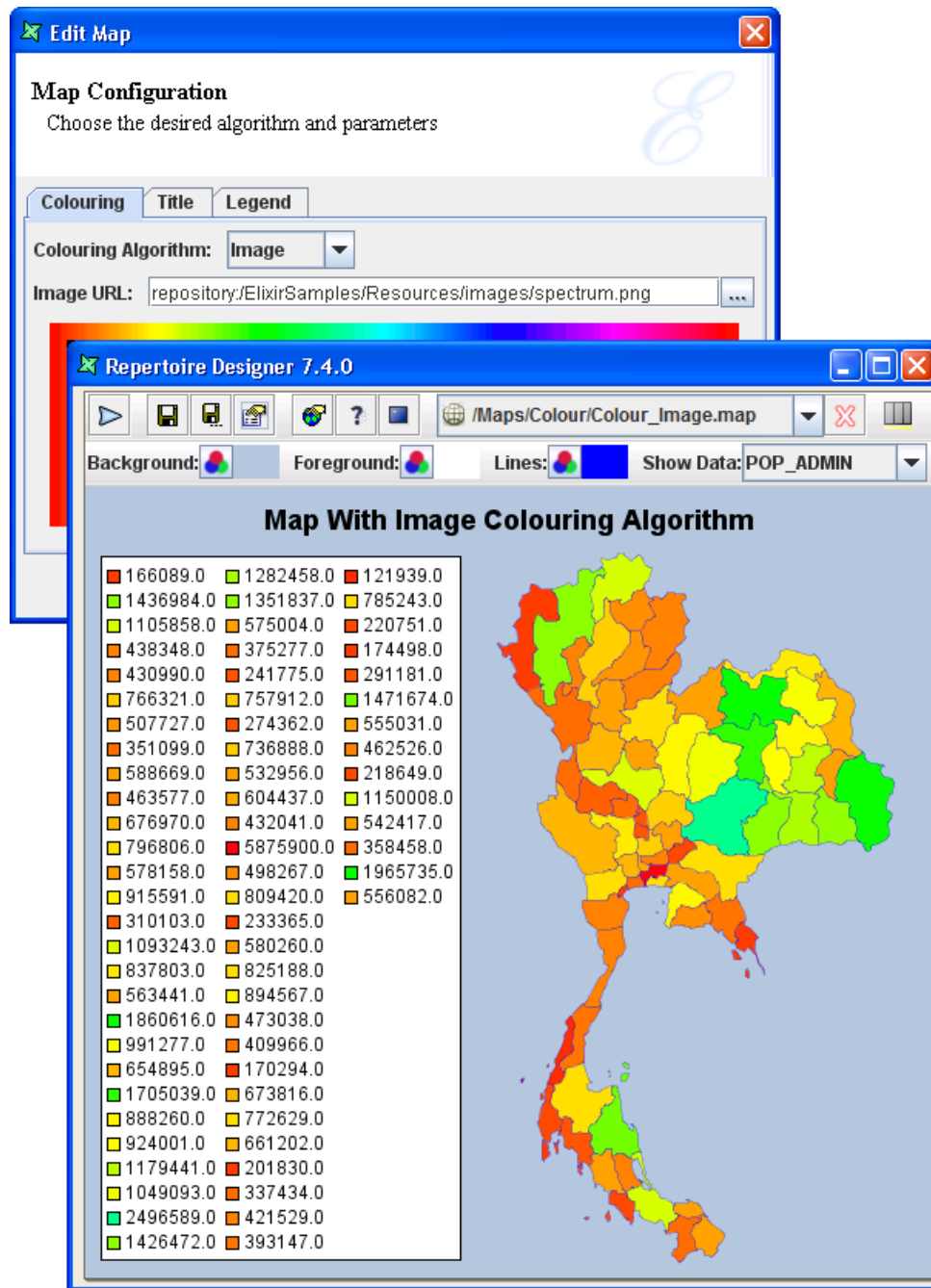
Figure 4.4. Colouring Algorithm - Gradient 3

Image

User can create their own images to choose exactly the gradient that they want. For example, user can create a “4-gradient” image with a dark green to medium green to light blue and then to white. Before using this option, ensure that the image file is available in the Repository.

In the Map Configuration wizard, in Colouring tab, select Image from the drop down list. Enter the URL of the image in Image URL: field.

Figure 4.5. Colouring Algorithm - Image



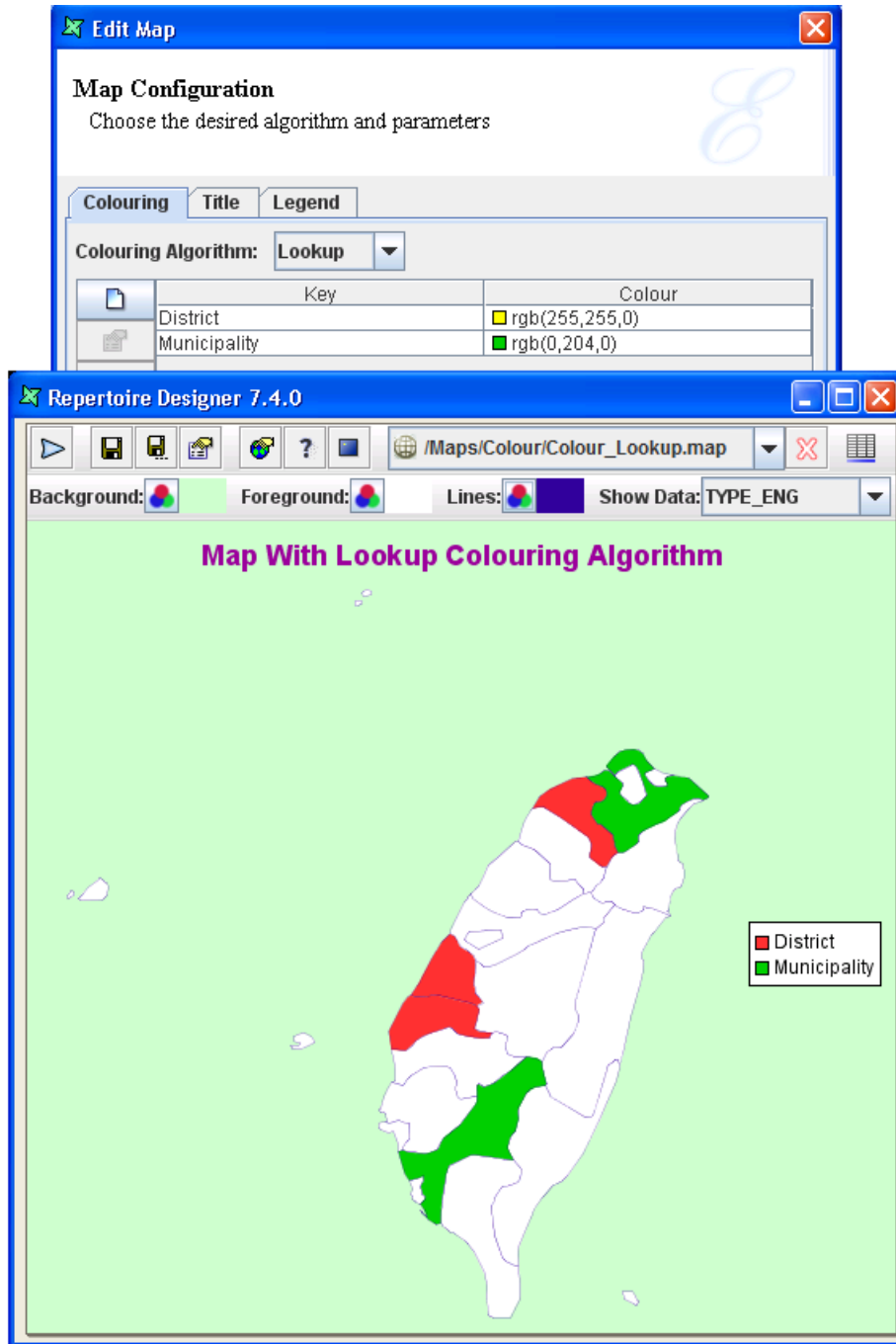
Tip

- If user requires each region to have its own unique colour, then a spectrum image which has a wide range of blended colours will be able to differentiate all the individual regions.

Lookup

- It is useful if user only wants some of the regions to appear coloured and the rest can follow the foreground colour of the base map. In Figure 4.1, “Colouring Algorithm - Array (with few regions)”, all categories are coloured. Thus if the user only needs a few particular categories to be coloured, this option can be used.

Figure 4.6. Colouring Algorithm - Lookup



Note

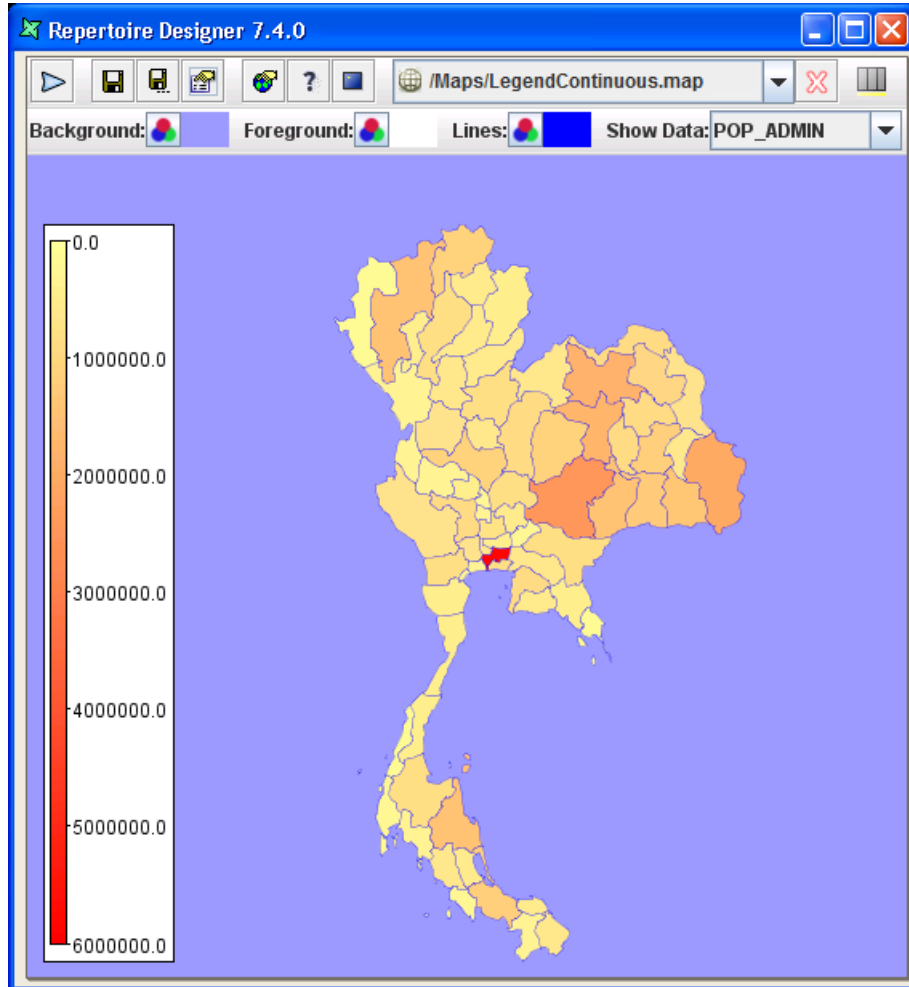
The labels entered for each colour specified under the key option must tally with the data of the particular region's field name required.

Legend

Legend will appear as a box on the map, to show the contents (in the form of *text* or *continuous range* format) of all the regions.

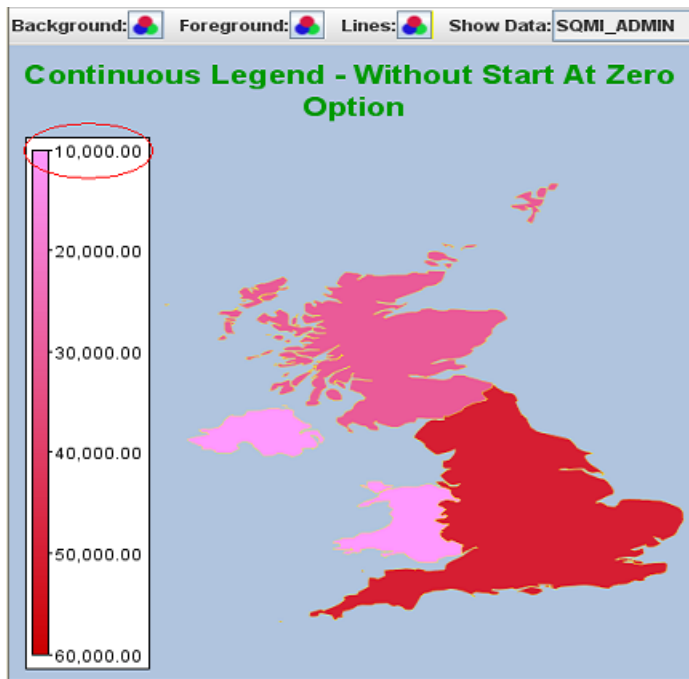
In the case of Figure 4.3, “Colouring Algorithm - Gradient 2”, the population size of all the regions are shown in the Legend box. Therefore, the map might not look presentable. By enabling the *Continuous* option, the legend becomes smaller and shows only a range of blended colours with default major tabs.

Figure 4.7. A Sample of Continuous Legend



Start At Zero option is a preference whether to show the range from zero or from the a rounded up figure nearest to the minimum and maximum 'fill-regions' data.

Figure 4.8. A Sample of Continuous Legend without Start At Zero Option



Note

- The *Continuous* and *Start At Zero* option can only be used when you select a numeric field name.
- The *Start At Zero* option will only be effective when the *Continuous* option is enabled too.

Map Layers

They are the add-on maps over the base map (the first map that user wants to appear as the outline first, as shown in Figure 4.9, “Example of Base Map without Map Layer”).

These add-on maps can be additional graphics like canals, road routes, shopping malls, libraries, etc.

From Figure 4.9, “Example of Base Map without Map Layer”, the blue strips are the Singapore canals, the black line represent the major highways and the red dots represent the hospitals and polyclinics whereby you can see the specific names when you hover the cursor over the red dots.

Figure 4.9. Example of Base Map without Map Layer

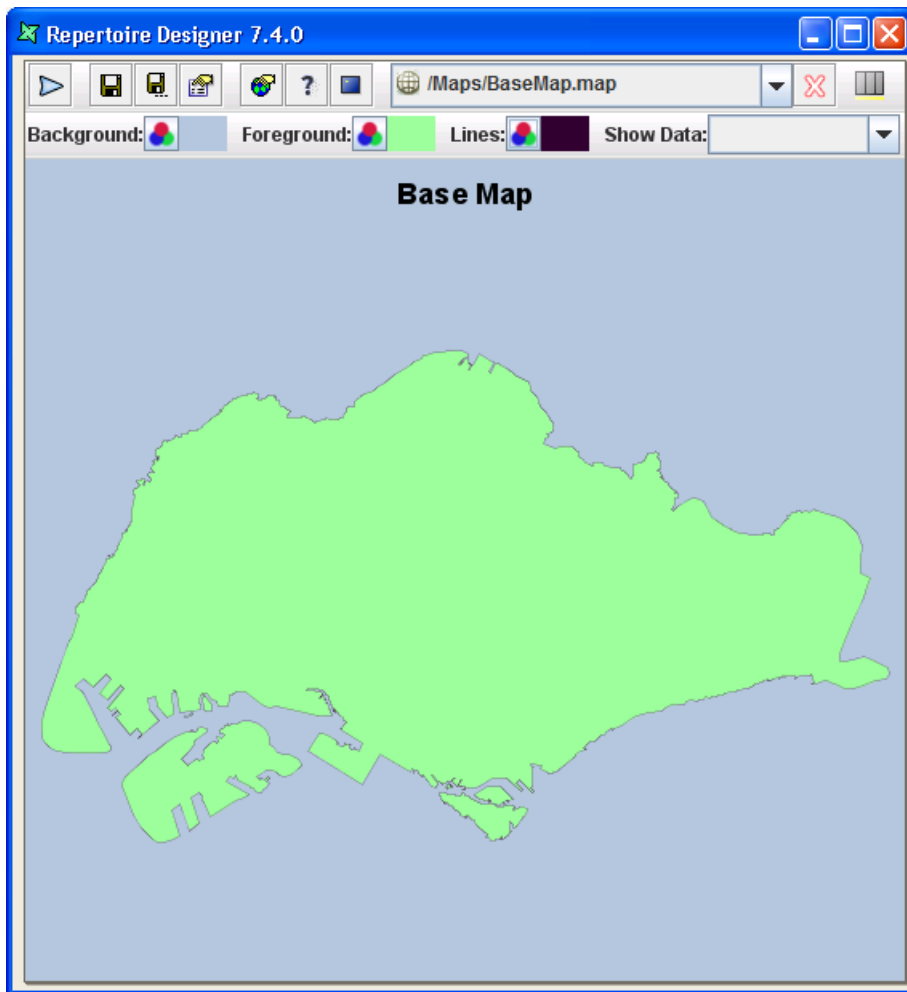
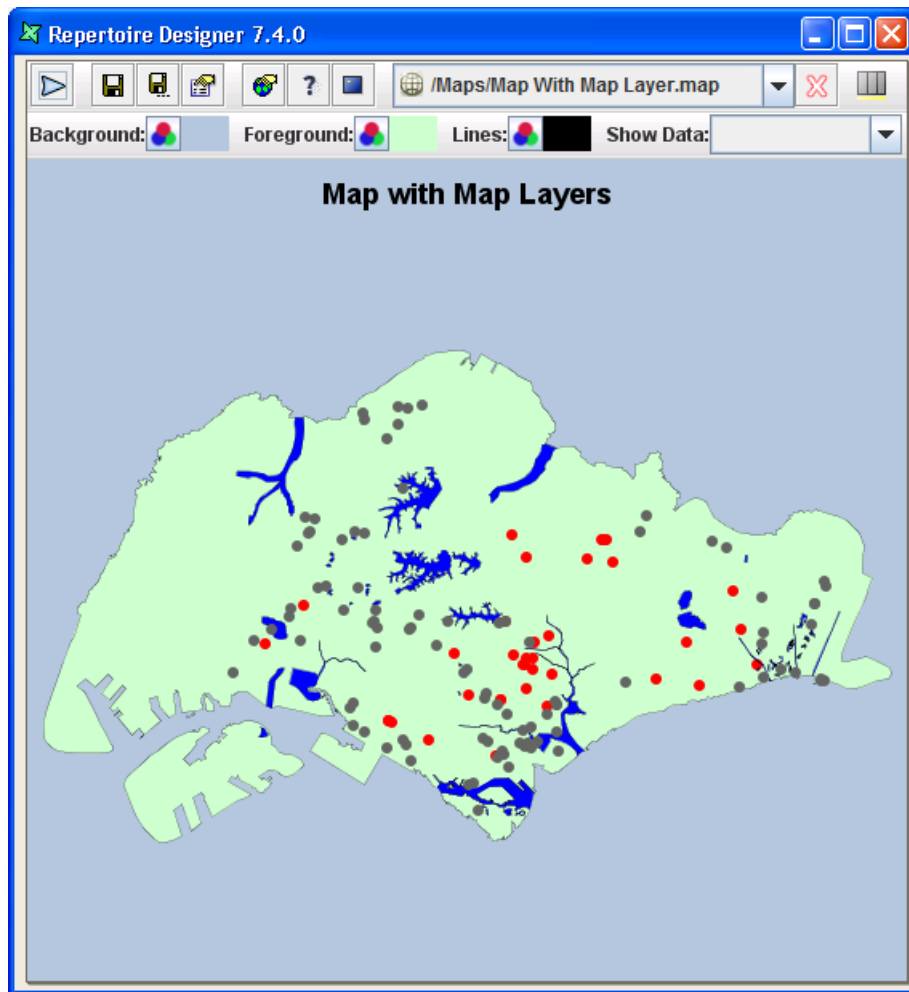


Figure 4.10. Example of a Base Map with Map Layers

Primary key

This key is a fieldname with non-duplicate data. It is necessary to select which key you intend to use as a primary key in order to show as meaningful “hovering legends” when you place your cursor over any region. By default, it will select the first fieldname (Column 1) as the primary key.