The PortResourceProxy

Table of content

Introduction

The PortResouceProxy is a small Proxy for local purposes. It was born from the idea to handle the "Same-Origin-Policy" by proxying all services that run on different ports. Therefore it is necessary to bunch all ports to deliver the specified resource. May there are some other fields of use, I doesn't mentioned The PortResourceProxy is really easy to configure and consists of just two files (Application and ResourceMap.xml). On github you'll get the code at a public repository. It is programmed in C# on .net 4.0 platform.

Feature List

- ability to proxy several port-services under on port
- free configurable source - destination "ResourceMap.xml" to set the http/resource pattern
- manual start and stop of the proxy
- console-like output of the status and errors

Status of the project (after initial release)

- there are some unreported bugs
- the keyfeatures are almost completly operative
- there are still some error messages and feedbacks, who has to add
- there are some Exceptions caused by the HttpListener - but they don't interrupt the operation
- may the proxy should get a better way of configuration
- just tested on Windows 7 (run as administrator)

First steps

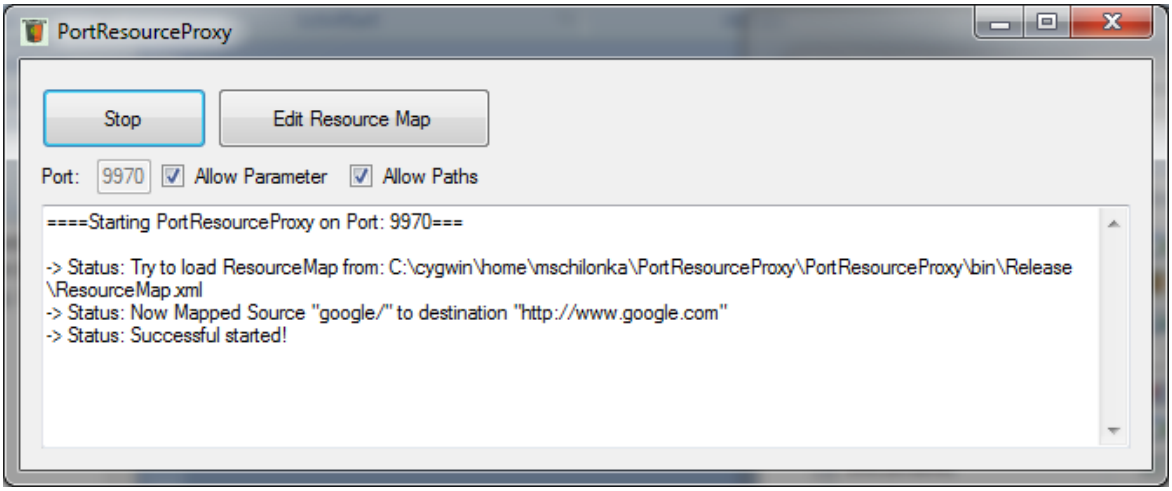
Set up the environment

To run the PortResourceProxy you'll need the current .net Framework. You can download it from here: http://www.microsoft.com/downloads/de-de/details.aspx?FamilyID=9CFB2D51-5FF4-4491-B0E5-B386F32C0992

Under Windows 7 you have to run this application as administrator by clicking the right mouse key → run as administrator. This is necessary because the PortResourceProxy opens a new port. There is no installation required.
Furthermore it is important, the ResourceMap.xml is located in the same folder like the Application. Take a glance on the example, if you don't know about the structure of the ResourceMap.xml. Let's shift attention how to edit the ResourceMap:

How to edit the ResourceMap

On the proxy monitor you've got a button named "Edit Resource Map". This one opens the notepad with the ResourceMap.xml. Here you CAN edit the Map. Of course, you could edit the file with your favorite xml editor.



After you clicked in "Start", the proxy reads the current ResourceMap.xml and will work to these settings. In the console, you'll get all the mapped resources.

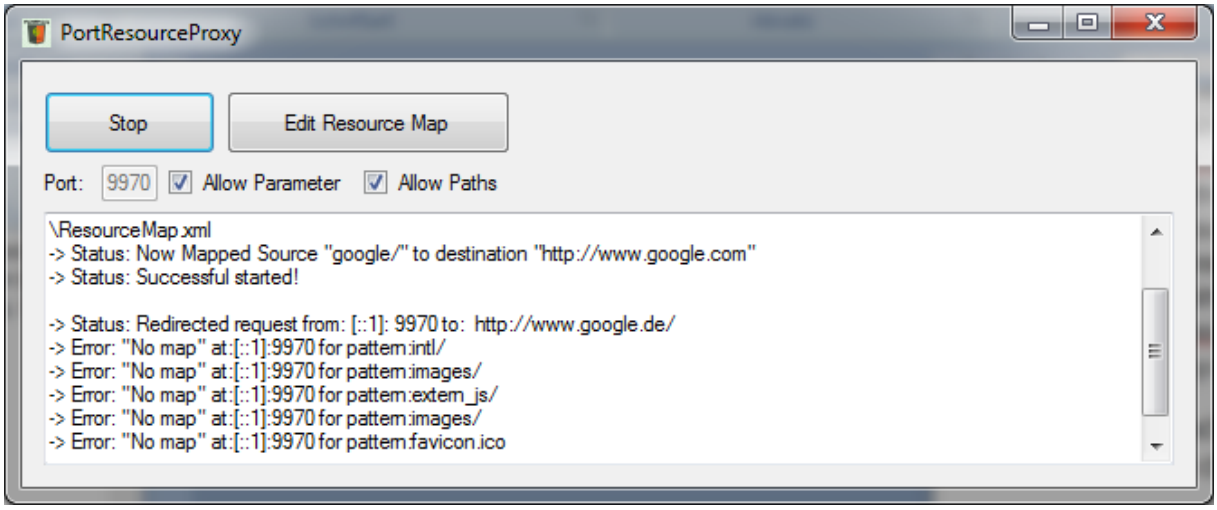
The base structure of the ResourceMap is like the following:

```
<?xml version="1.0" encoding="utf-8" ?>
 <Proxy>
  <ResourceMap>
    <Path>
      <source>google/</source>
      <destination>http://www.google.com</destination>
    </Path>
  </ResourceMap>>
 </Proxy>
```

For each path you want to proxy, you have to declare a "Path" node. A "Path" node consists of two children, source and destination.

In this example you got:

All activities will be shown in the console output:



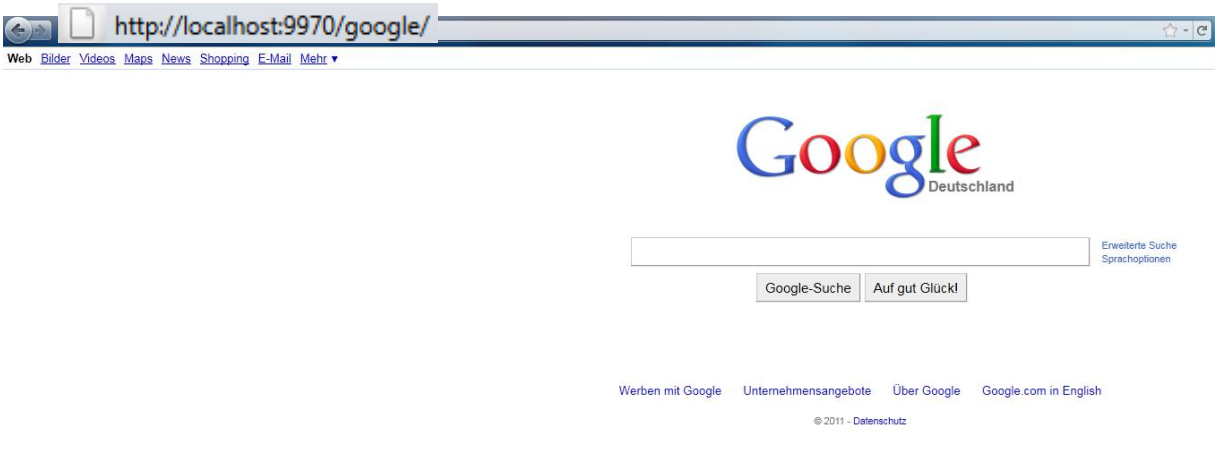
As you see, there was a succeeded redirection for “google/” to http://www.google.de
(Why google.de? Google does a geographic redirection by themselves).

But there are some erros, too. The browser now loads automatically some resources like javascript or images. They are sometimes at other locations. If you want, that this request are also successful, you have to set this up in ResourceMap, as well.

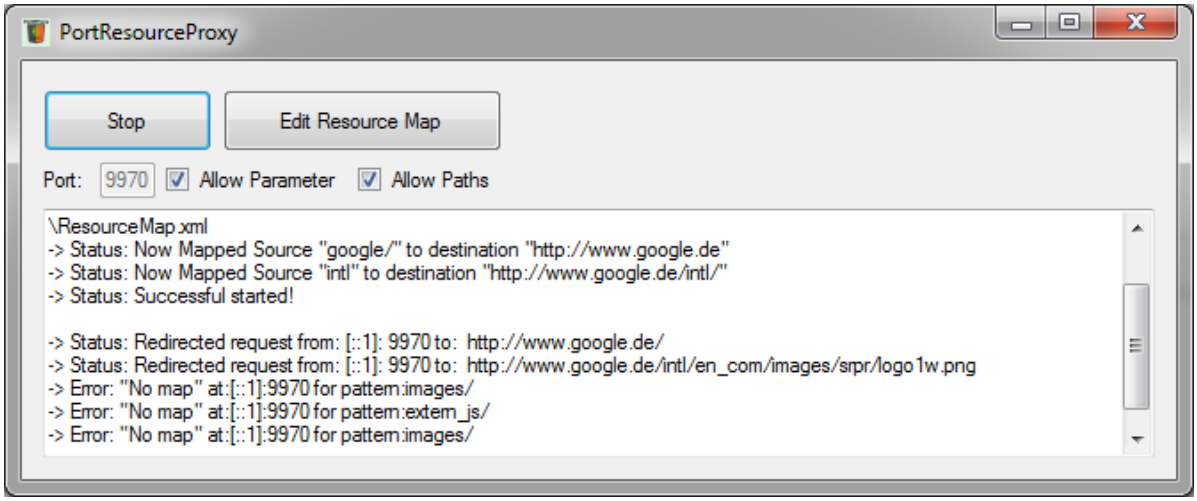
Ok. Let’s create a further map for the images of the google page.

```
<?xml version="1.0" encoding="utf-8" ?>
 <Proxy>
  <ResourceMap>
    <Path>
      <source>google/</source>
      <destination>http://www.google.de</destination>
    </Path>
    <Path>
      <source>intl/</source>
      <destination>http://www.google.de/intl/</destination>
    </Path>
  </ResourceMap>>
 </Proxy>
```

The result should be:



And the console output:



Every went fine ;)

IMPORTANT: Please make sure, that the ResourceMap contains a correct structure and is available in the same folder like the application to ensure a correct operation!!!

Usage

See: How to edit ResourceMap.

The proxy monitor is quite intuitive. Start/Stop with the expected button and set the port you want to use with the related text field.
There are two check boxes which allow / deny parameters and paths.
If you don't allow parameters, you can't request a resource with parameters through the proxy. Currently there is no feedback about it.
If you don't allow paths, you can't request resources, which are located under the root you declared. An origin path on the same level has to be declared as independent map.

There are a lot of "→ Status:" and "→ Error:" messages in the console, but each of them should be self explanatory.

Get involved at github

The complete source code is provided at: https://github.com/Schille/PortResourceProxy.
Feel free to fork or develop on this project.