

# LR Web Service 简单入门

云层 2011-8-23

正好有学生问到这个问题，而且准备在《性能测试进阶指南》第二版里面写这个东西，所以先随手写个简单入门给大家参考把。

这里我使用的是外网的一个 Web Service 服务，使用 LR11 作为开发工具。

网站地址 [http://www.webxml.com.cn/zh\\_cn/index.aspx](http://www.webxml.com.cn/zh_cn/index.aspx)

这里我使用的是该网站的天气预报服务

<http://webservice.webxml.com.cn/WebServices/WeatherWS.asmx>

在使用 web service 前，最关键的就是服务说明了，连接 web service 其实有好几种方法，包括 http, soap 等

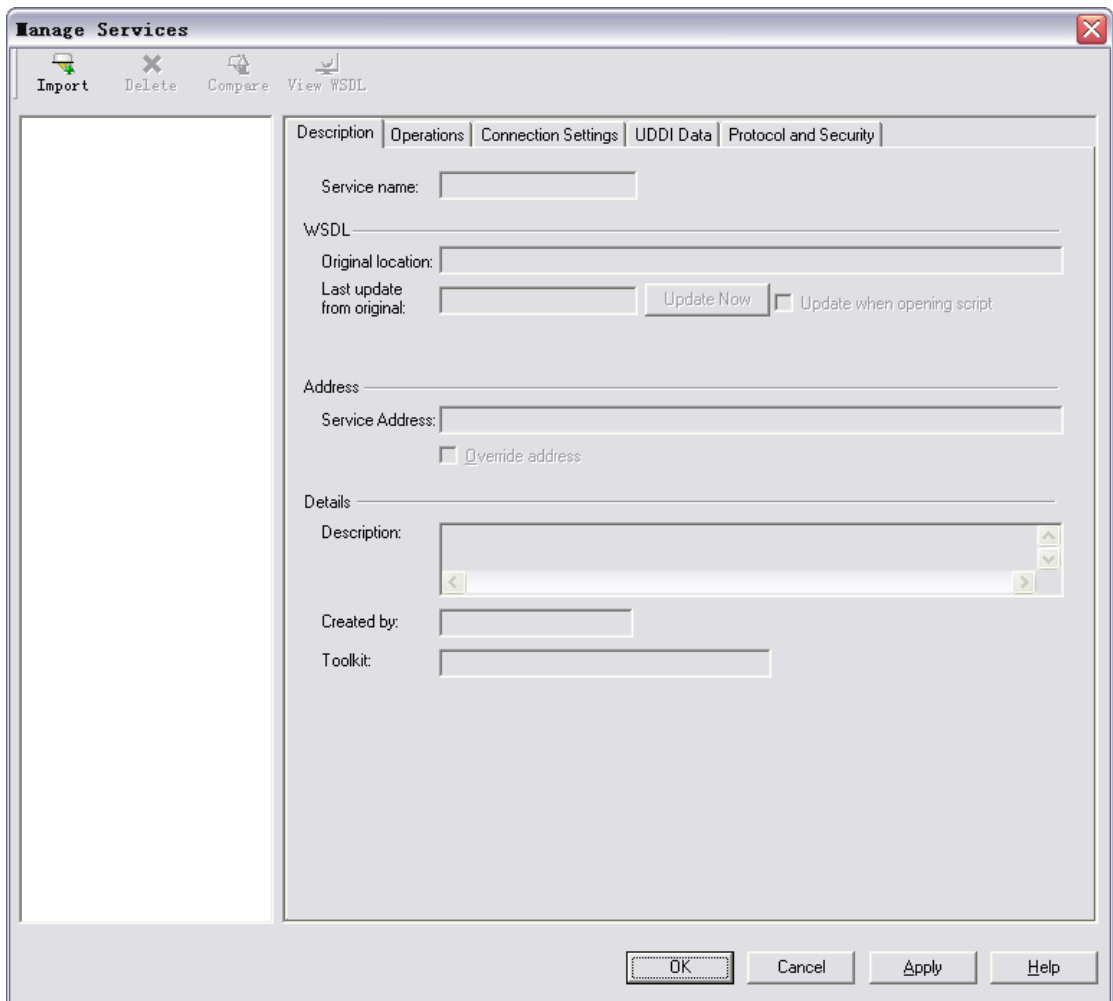
启动 LR 选择 web service 协议

整个过程的关键就是下面这几个东西

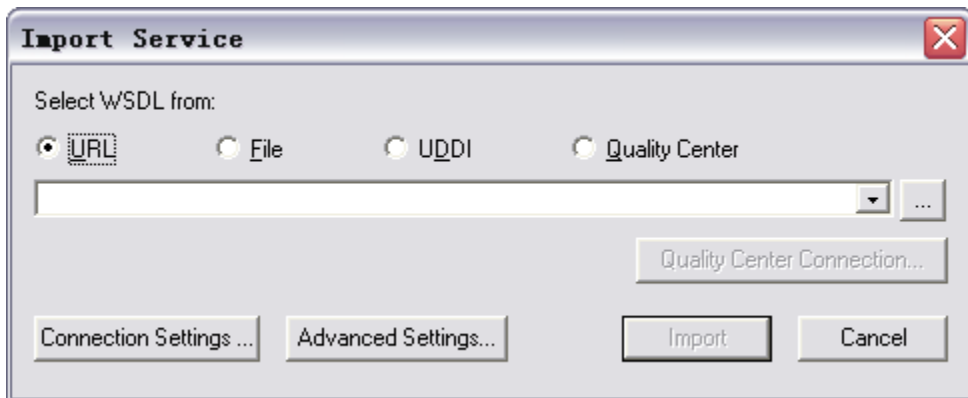


导入 SOAP,管理服务 and 添加服务调用，刚开始大家会觉得 web service 好难，最后你会发现其实 web service 是最简单的，比 HTML 简单多了

1. 首先我们需要添加一个服务调用，点击 manage services



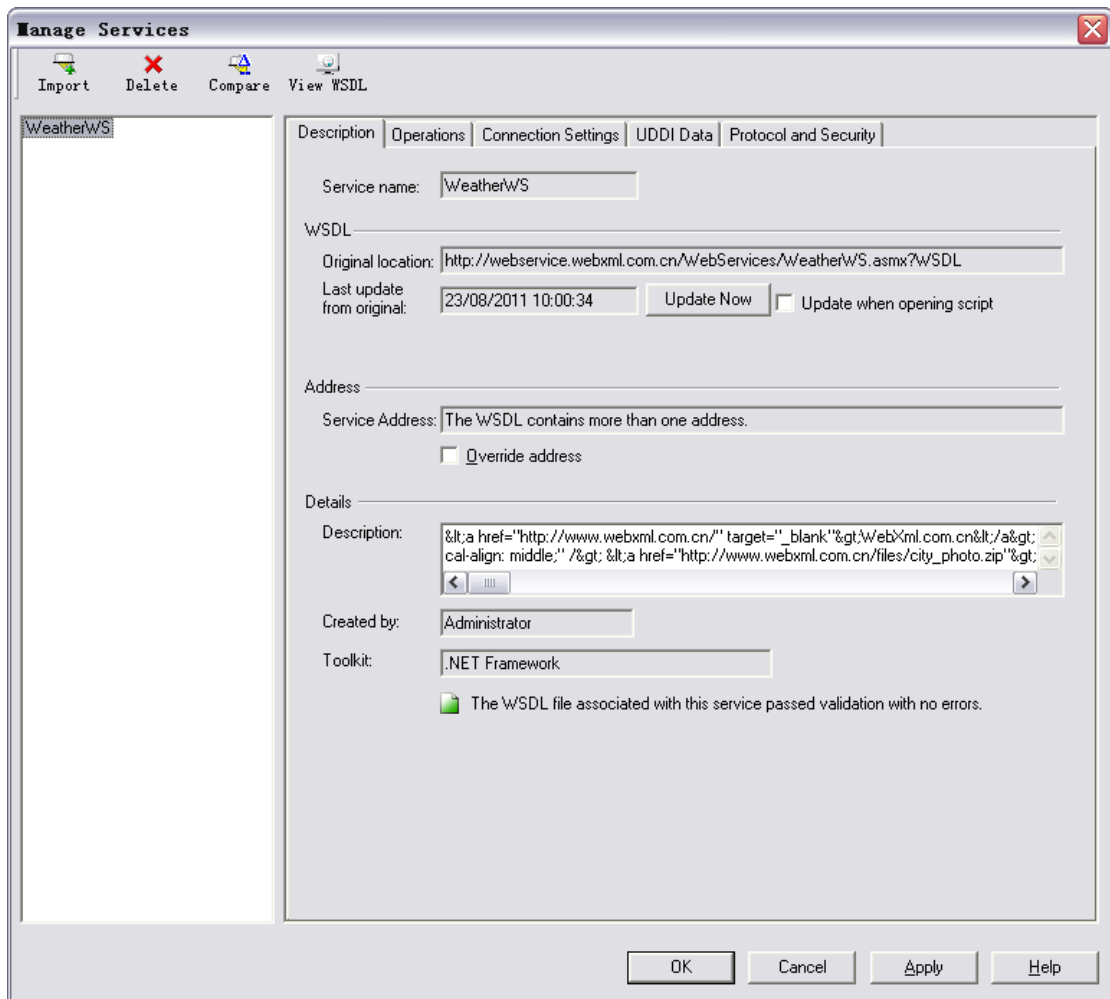
在这里我们选择 Import 导入连接方式



这里我们使用的连接方式是 URL，那么应该写啥呢，写 Web Service 中的 WSDL 地址，该地址为

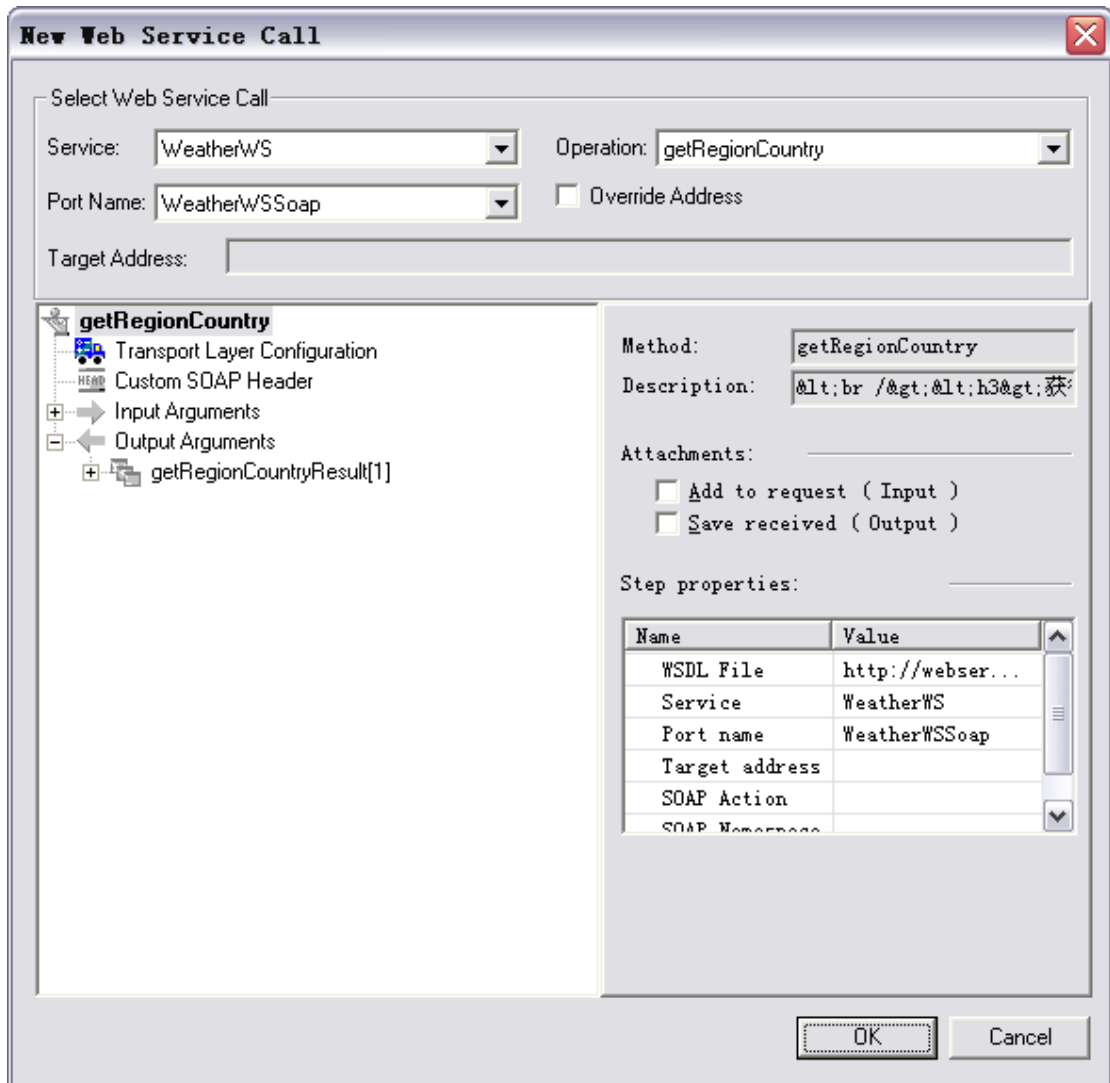
<http://webservice.webxml.com.cn/WebServices/WeatherWS.asmx?WSDL>

就是服务页面上有个服务说明的链接，如果是自己开发那么可以询问一下开发这个东西在那里，如果这东西没有那么 webservice 就没法用了，输入地址后导入

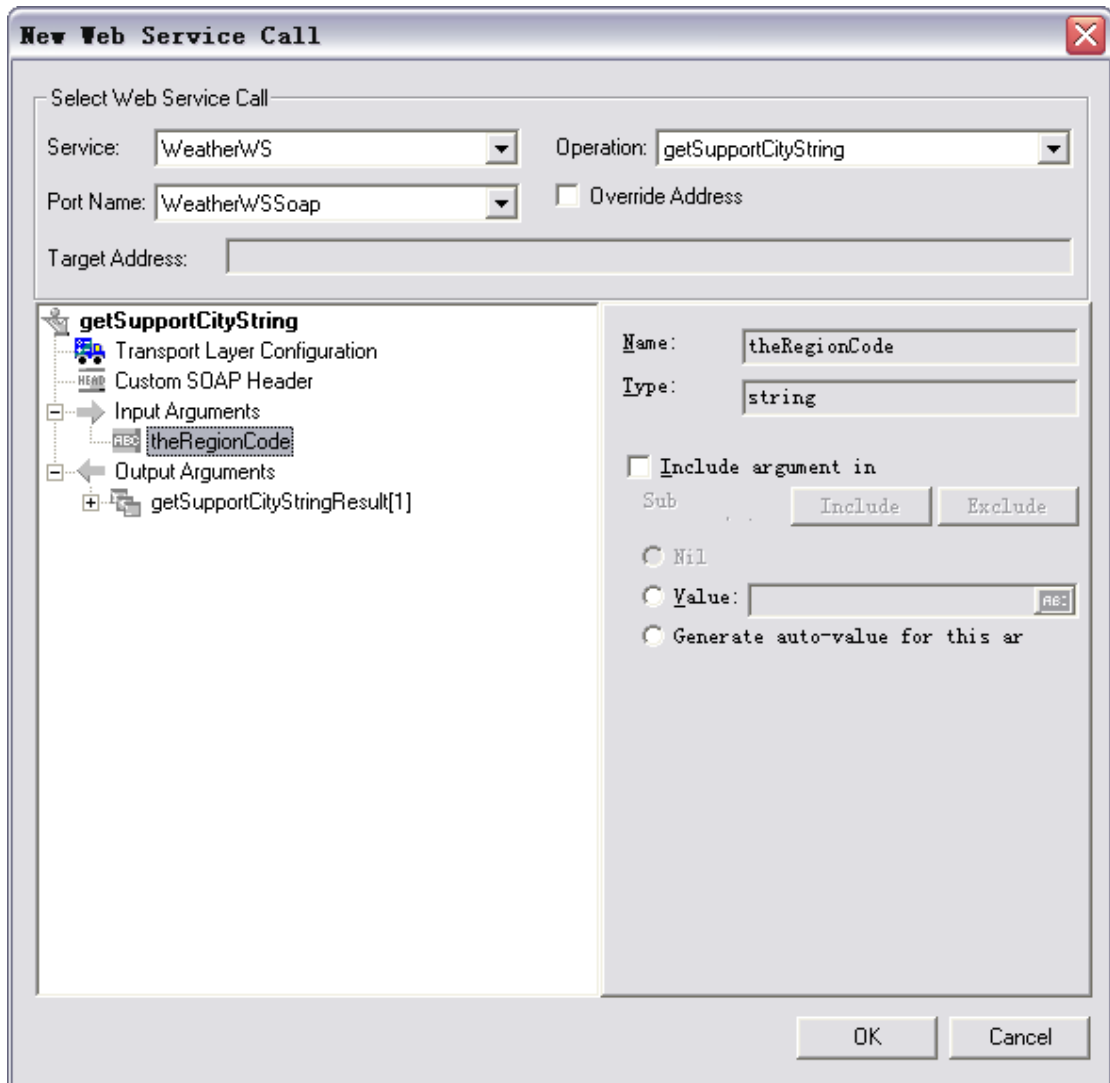


这个时候你会发现整个服务都被导入了，这就是我们要做的第一步。

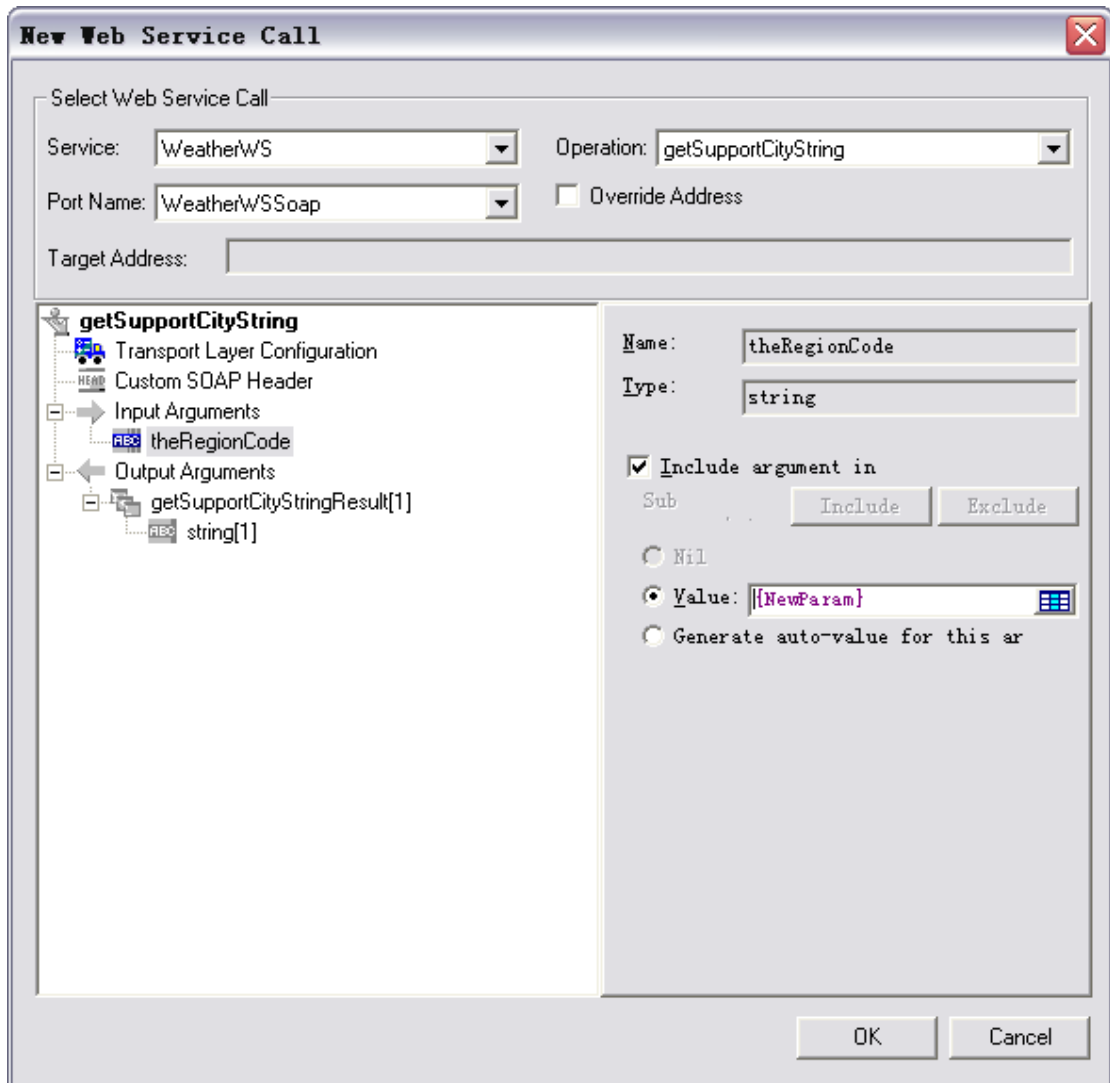
2. 其次我们要开始写 service call 服务调用，其实就和写方法调用没啥区别



点击后会看到上面这个界面出来，这里右侧列出了该服务上得所有方法，左侧列出了该方法的输入输出参数  
首先我们来调用一下城市查询，了解每个城市对应的编号主键



这里我们选择 `getSupportCityString` 方法来做，这里面带一个输入，一个字符串输出。我们对输入 `theRegionCode` 做一个简单的参数化，参数值为“上海”



去 param list 配置 NewParam 这个参数的值为“上海”，得到代码

```
Action()
{
```

```
    web_service_call( "StepName=getSupportCityString_101",
        "SOAPMethod=WeatherWS|WeatherWSSoap|getSupportCityString",
        "ResponseParam=response",
        "Service=WeatherWS",
        "ExpectedResponse=SoapResult",
        "Snapshot=t1314065114.inf",
        BEGIN_ARGUMENTS,
        "theRegionCode={NewParam}",
        END_ARGUMENTS,
        BEGIN_RESULT,
        END_RESULT,
        LAST);
```

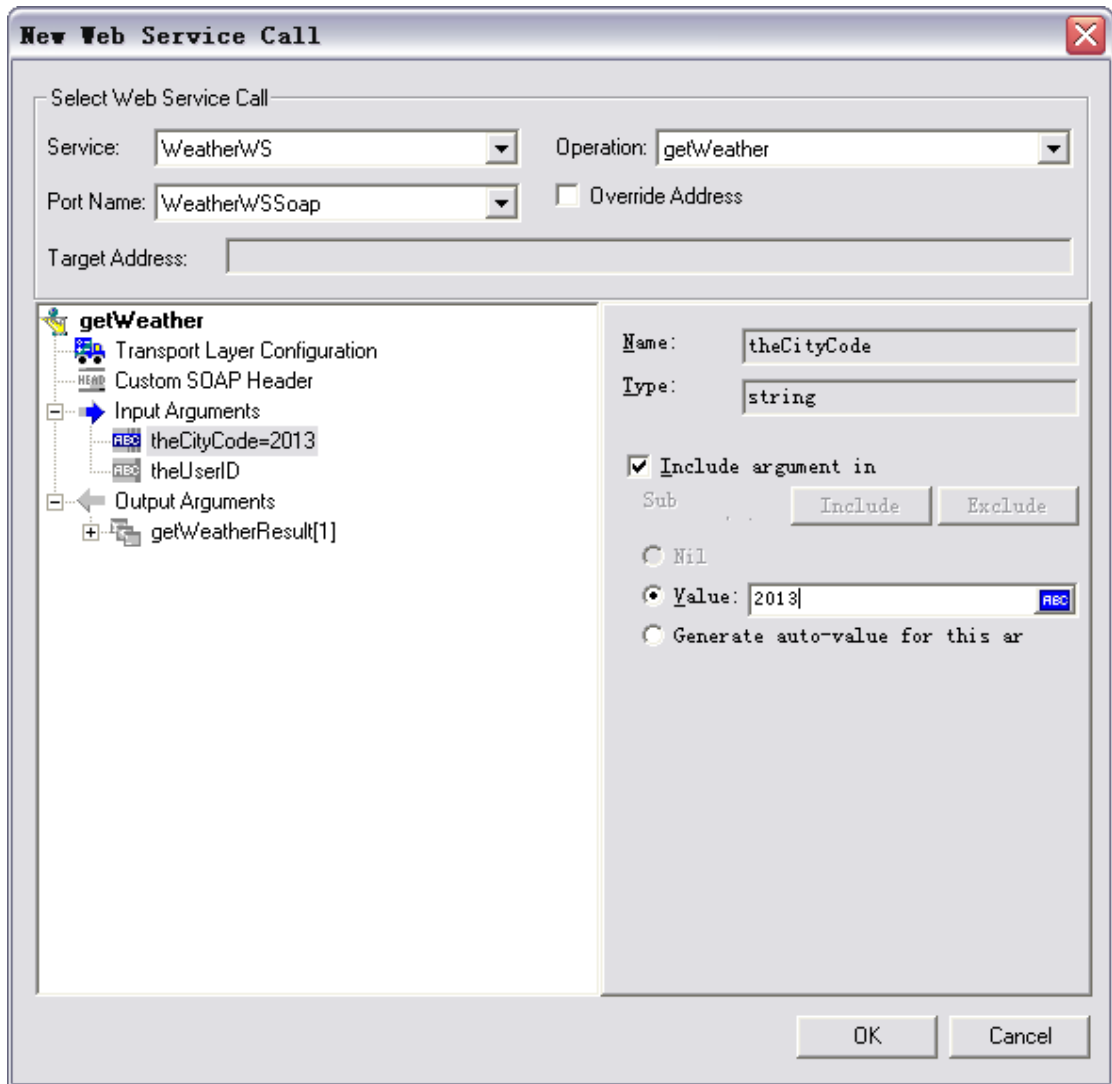
```
return 0;
}
```

这个代码就是调用 web service 的代码，在运行之前打开参数取值日志，运行脚本就可以看到效果了。日志如下：

```
Virtual User Script started at : 2011-08-23 10:06:13
Starting action vuser_init.
Web Services replay version 11.0.0 for WIN2003; Toolkit: ".Net"; build 8859
Run-Time Settings file: "C:\Documents and Settings\Administrator\Local Settings\Temp\noname1\default.cfg"
Vuser directory: "C:\Documents and Settings\Administrator\Local Settings\Temp\noname1"
Vuser output directory: "C:\Documents and Settings\Administrator\Local Settings\Temp\noname1\"
LOCAL start date/time: 2011-08-23 10:06:14
Ending action vuser_init.
Running Vuser...
Starting iteration 1.
Starting action Action.
Action.c(4): Notify: Parameter Substitution: parameter "NewParam" = "上海"
Action.c(4): Web service call "getSupportCityString_101" started
Action.c(4): Notify: Saving Parameter "getSupportCityString_101_Response" =
<getSupportCityString><getSupportCityStringResult><string> 宝 山 ,2009</string><string> 崇
明,2012</string><string> 奉贤,2063</string><string> 嘉定,2011</string><string> 金山,3530</string><string> 闵
行,2008</string><string> 南汇,2014</string><string> 浦东,2015</string><string> 青浦,2061</string><string> 上
海 ,2013</string><string> 松 江 ,3413</string><string> 徐 家
汇,3643</string></getSupportCityStringResult></getSupportCityString>".
Action.c(4): Notify: Saving Parameter "response = <?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Body><getSupportCityStringResponse
xmlns="http://WebXml.com.cn/"><getSupportCityStringResult><string> 淪 濱 北 ,2009</string><string> 宕 困
霖 ,2012</string><string> 濂 夔 播 ,2063</string><string> 鎬 文 畚 ,2011</string><string> 閱 賤
北 ,3530</string><string> 闊 佃 ,2008</string><string> 錡 榕 朕 ,2014</string><string> 嫻 一
篔 ,2015</string><string> 闌 椒 鄆 ,2061</string><string> 涓 嫫 搗 ,2013</string><string> 鏃 炬
曠 ,3413</string><string> 寰 愬 姪
\x87,3643</string></getSupportCityStringResult></getSupportCityStringResponse></soap:Body></soap:Envelop
e>".
Action.c(4): Web service call "getSupportCityString_101" was successful
Ending action Action.
Ending iteration 1.
Ending Vuser...
Starting action vuser_end.
Ending action vuser_end.
Vuser Terminated.
```

通过返回我们可以看到上海的编号是 2013

接着我们可以继续通过这个值调用下一个服务了，GetWeather 调用天气预报，同样方式加载 CALL



这里我们先不考虑怎么关联得到上海的编号叫做 2013 直接写，然后 theUserID 这里可以不填，直接确定完成该服务的调用，再次运行，上海的天气就出来了。

Action.c(18): Notify: Saving Parameter "getWeather\_101\_Response = <getWeather><getWeatherResult><string>直辖市 上海</string><string>上海</string><string>2013</string><string>2011/08/23 09:55:50</string><string>今日天气实况: 气温: 25°C; 风向/风力: 东风 1 级; 湿度: 73%</string><string>空气质量: 良; 紫外线强度: 弱</string><string>穿衣指数: 天气炎热, 建议着短衫、短裙、短裤、薄型 T 恤衫、敞领短袖棉衫等清凉夏季服装。\\n 感冒指数: 各项气象条件适宜, 无明显降温过程, 发生感冒机率较低。\\n 运动指数: 有降水, 风力较强, 可适当在户内轻度运动。若坚持户外运动, 选择合适运动, 并携带雨具。\\n 洗车指数: 不宜洗车, 未来 24 小时内有雨, 如果在此期间洗车, 雨水和路上的泥水可能会再次弄脏您的爱车。\\n 晾晒指数: 偶尔的降雨可能会淋湿晾晒的衣物, 不太适宜晾晒。请随时注意天气变化。\\n 旅游指数: 有阵雨, 温度适宜, 在细雨中游玩别有一番情调, 可不要错过机会呦! 但记得出门要携带雨具。\\n 路况指数: 有小雨, 路面潮湿, 车辆易打滑, 请小心驾驶。\\n 舒适度指数: 白天有降雨, 但会使人们感觉有些热, 不过大部分人仍会有比较舒适的感觉。\\n 空气污染指数: 气象条件有利于空气污染物稀释、扩散和清除, 可在室外正常活动。\\n 紫外线指数: 紫外线强度较弱, 建议出门前涂擦 SPF 在 12-15 之间、PA+ 的防晒护肤品。</string><string>8 月 23 日 阵雨 </string><string>24 °C /28 °C </string><string>东北风 3-4 级</string><string>3.gif</string><string>3.gif</string><string>8 月 24 日 阵雨 </string><string>24 °C /28 °C



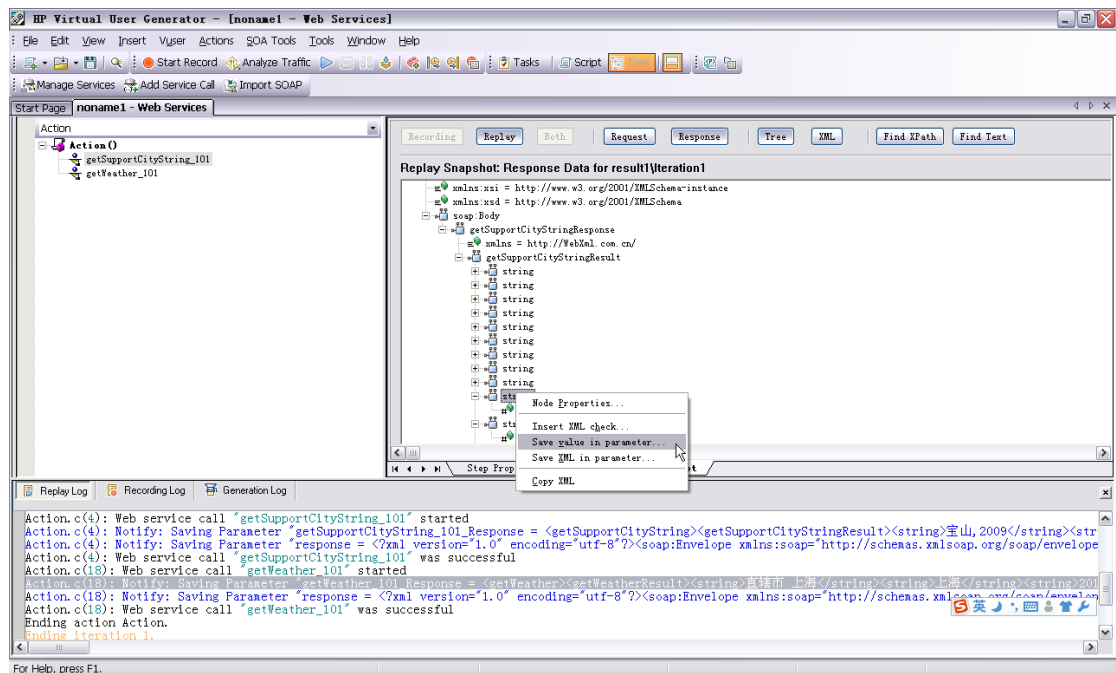
```
</string><string>东北风 3-4 级</string><string>3.gif</string><string>3.gif</string><string>8 月 25 日 阵雨</string><string>24 °C /28 °C </string><string> 东北风 3-4 级</string><string>3.gif</string><string>3.gif</string><string>8 月 26 日 阵雨</string><string>25 °C /28 °C</string><string>东北风 3-4 级</string><string>3.gif</string><string>3.gif</string><string>8 月 27 日 阵雨</string><string>24 °C /28 °C </string><string> 东北风 3-4 级</string><string>3.gif</string><string>3.gif</string></getWeatherResult></getWeather>".
```

3. 关联返回，当服务调用后我们可以需要对返回做获得并且支持后续操作，例如前面的城市编号，或者某些 Sessionid

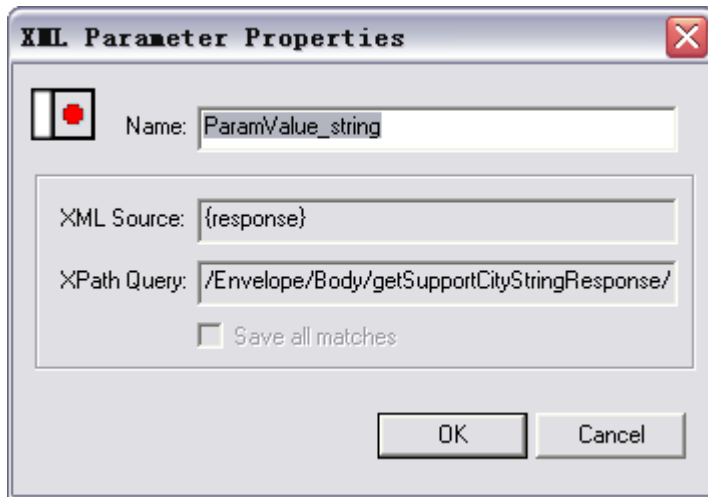
其实我们在日志里面已经可以看到系统自动帮我们把所有内容都“关联”到一个参数里面去了，问题只是在怎么能够把里面的值获得，这里要用到的函数是 `lr_get_xml_value`，通过这个函数可以把 XML 结构的内容按照某个节点的方式获得。

直接写可能不太好写出节点层次，这里可以使用图形化方式来做

把脚本切到 `Tree` 模式，选择服务的 `snapshot`，只有脚本回访后才有，选择 `Response` 请求应答，我们就可以看到一个完整的返回 XML 格式，找到自己好的数据段，我这里选择是上海这个节点，右键 `save value in parameter`



系统会自动帮你识别节点结构



我们只需要确认就行了，代码生成，回访检查关联结果  
`lr_xml_get_values("XML={response}",`

```
"FastQuery=/Envelope/Body/getSupportCityStringResponse/getSupportCityStringResult/string[10]",
    "ValueParam=ParamValue_string",
    LAST);
```

运行结果如下

```
Action.c(18): Notify: Saving Parameter "ParamValue_string = 上海,2013".
Action.c(18): "lr_xml_get_values" succeeded, 1 match processed
```

这样我们就关联得到了该节点的内容了，这个节点有些麻烦，包含两个属性，那么由于编号是固定的 4 位长度，可以考虑使用 `lr_save_var` 或者 `strtok` 来解决（这个东西自己参考我第一版里面的章节了，不想在这里再写了）

这是单一关联如果要多关联怎么办？  
 那么这样做，将上面的代码修改为！

```
int i;
web_service_call( "StepName=getSupportCityString_101",
    "SOAPMethod=WeatherWS|WeatherWSSoap|getSupportCityString",
    "ResponseParam=response",
    "Service=WeatherWS",
    "ExpectedResponse=SoapResult",
    "Snapshot=t1314065114.inf",
    BEGIN_ARGUMENTS,
    "theRegionCode={NewParam}",
    END_ARGUMENTS,
    BEGIN_RESULT,
    END_RESULT,
    LAST);
```

```
i=Lr_xml_get_values("XML={response}",  
  
"Query=/Envelope/Body/getSupportCityStringResponse/getSupportCityStringResult/string",  
    "ValueParam=ParamValue_string",  
    "SelectAll=yes",  
    LAST);
```

注意变化，多了一个 SelectAll=yes 的数据段，Query 也不是 FastQuery 了，而且查询的 XML 节点也变了，前面还多了个整数变量 i

再次运行，得到日志

```
Action.c(18): Notify: Saving Parameter "ParamValue_string_1 = 宝山,2009".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_2 = 崇明,2012".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_3 = 奉贤,2063".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_4 = 嘉定,2011".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_5 = 金山,3530".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_6 = 闵行,2008".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_7 = 南汇,2014".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_8 = 浦东,2015".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_9 = 青浦,2061".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_10 = 上海,2013".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_11 = 松江,3413".  
Action.c(18): Notify: Saving Parameter "ParamValue_string_12 = 徐家汇,3643".
```

看到这里很简单了把，最后要做的就是

```
Lr_save_int(i,ParamValue_string_count);
```

这样就得到了参数数组 ParamValue\_string，剩下的就不用多说了把。