

---

# **MapGIS K9 IMS FLEX 问题集锦**

---

# 目 录

## 第一章 开发入门篇

第 1 问	MapGIS K9 IMS 需要什么样的开发环境 .....	1
第 2 问	怎样配置 MapGIS K9 IMS 开发环境 .....	1
第 3 问	怎样配置 Flex 开发环境 .....	5
第 4 问	怎样创建一个 MapGIS K9 IMS Flex 项目 .....	5
第 5 问	怎样使用 MapGIS K9 IMS Flex 功能组件 .....	10
第 6 问	怎样更新 MapGIS K9 IMS Flex 功能组件 .....	13

## 第二章 地图显示篇

第 1 问	怎样使用地图容器 .....	16
控件式:	.....	16
代码式:	.....	16
第 2 问	怎样显示瓦片地图 .....	17
控件式:	.....	17
代码式:	.....	17
第 3 问	怎样显示地图文档 .....	18
控件式:	.....	18
代码式:	.....	19
第 4 问	怎样叠加显示瓦片地图和地图文档 .....	19
控件式:	.....	19
代码式:	.....	20
第 5 问	怎样同时显示多个地图窗口 .....	21
控件式:	.....	21
代码式:	.....	22
第 6 问	怎样添加一张图片跟地图叠加显示 .....	23
第 7 问	怎样去除标注弹出窗口的图片 .....	24
第 8 问	怎样使地图容器与鹰眼显示不同地图 .....	24
控件式:	.....	24
代码式:	.....	25
第 9 问	怎样切换鹰眼显示和隐藏状态 .....	26
第 10 问	怎样实现不同的客户端看到的地图不同 .....	27
第 11 问	怎样实现鼠标点击或移到某个对象上获取某对象的属性 .....	28
第 12 问	什么原因会引起地图不能正常显示 .....	30
第 13 问	什么原因导致本机可以出图但其他机器访问不能出图 .....	30
第 14 问	怎样在网页中嵌入 MapGIS K9 IMS Flex 应用 .....	30

## 第三章 地图控制篇

第 1 问	怎样复位地图.....	31
第 2 问	怎样刷新地图.....	32
第 3 问	怎样控制地图显示级别.....	32
第 4 问	怎样使地图在某一显示级别居中显示某个位置.....	33
第 5 问	怎样移动地图至某一位置.....	34
第 6 问	怎样控制地图窗器中的图层是否显示.....	35

## 第四章 图形绘制篇

第 1 问	怎样在地图上绘制点.....	36
第 2 问	怎样在地图上绘制线.....	38
第 3 问	怎样在地图上绘制圆.....	39
第 4 问	怎样在地图上绘制矩形.....	41
第 5 问	怎样在地图上绘制区.....	42
第 6 问	怎样在地图上添加一个点.....	44
第 7 问	怎样在地图上添加一条线.....	45
第 8 问	怎样在地图上添加一个标注点.....	46
第 9 问	怎样实现地图窗口坐标和逻辑坐标之间的转换.....	47
第 10 问	怎样将统计图显示在地图上并跟随地图放大缩小.....	48
第 11 问	怎样添加一个热点区域.....	49

## 第五章 开发函数篇

第 1 问	怎样获取地图文档图层列表.....	50
第 2 问	怎样更新地图文档中的图层的状态.....	51
第 3 问	怎样获取鼠标当前位置信息.....	52
第 4 问	怎样添加和删除地图鼠标事件.....	53
第 5 问	怎样用鼠标获取一个点.....	54
第 6 问	怎样用鼠标获取一条线.....	55
第 7 问	怎样用鼠标获取一个矩形.....	57
第 8 问	怎样用鼠标获取一个圆.....	58
第 9 问	怎样用鼠标获取一个多边形.....	59
第 10 问	怎样实现点击查询.....	60
第 11 问	怎样实现线查询.....	62
第 12 问	怎样实现矩形查询.....	64
第 13 问	怎样实现圆查询.....	66
第 14 问	怎样实现多边形查询.....	68
第 15 问	怎样实现属性条件查询.....	71
第 16 问	怎样实现点击+条件查询.....	72
第 17 问	怎样实现线+条件查询.....	74
第 18 问	怎样实现矩形+条件查询.....	76
第 19 问	怎样实现圆+条件查询.....	78
第 20 问	怎样实现多边形+条件查询.....	81

第 21 问	怎样添加一个要素.....	83
第 22 问	怎样编辑一个要素.....	84
第 23 问	怎样删除一个要素.....	86
第 24 问	怎样对一个要素做缓冲区分析.....	89
第 25 问	怎样获取要素空间坐标信息并在地图上闪烁.....	91
第 26 问	怎样获取要素图形参数.....	94
第 27 问	怎样更新要素图形参数.....	96
第 28 问	怎样实现圆裁剪.....	99
第 29 问	怎样实现多边形裁剪.....	101
第 30 问	怎样实现拓扑分析.....	103
第 31 问	怎样实现路径分析.....	104
第 32 问	怎样获取 MapGIS 数据源列表.....	105
第 33 问	怎样获取 MapGIS 数据源中的数据库列表.....	106
第 34 问	怎样获取 MapGIS 数据库中的图层列表.....	107
第 35 问	怎样通过图层索引获取图层名称.....	109
第 36 问	怎样实现两个图层叠加分析.....	110
第 37 问	怎样在操作结束后把鼠标切换回空状态.....	111

## 第六章 高级篇

第 1 问	怎样实现自定义控件开发.....	112
第 2 问	怎样实现与 Google Map 的融合.....	114
第 3 问	怎样实现负载均衡.....	115
第 4 问	怎样使用 Web 转发器转发 Flex 请求到 GIS 服务器.....	116

# 第一章 开发入门篇

## 第1问 MapGIS K9 IMS 需要什么样的开发环境

MapGIS K9 IMS 是新一代的网络 GIS 开发平台，提供丰富多样的二次开发方式，包括富客户端的 Flex 开发。

基于 Flex 的 MapGIS K9 IMS 开发环境，包括操作系统、MapGIS K9 IMS 平台、数据库、浏览器、Flex 集成开发环境。

下面主要介绍在 Windows 系统下的开发环境与相关工具：

- **操作系统：**Windows 系列，包括 Windows 2000、Windows XP、Windows 2003、Windows 2008 等系列版本。推荐使用 Windows Server 2003 【SP2】，该版本性能稳定，适用于做开发。
- **.NET 环境：**Microsoft Visual Studio .NET 2005（.NET Framework 2.0 等）
- **Web 服务器：**Microsoft Internet Information Server 5.0(IIS5.0)或更高版本
- **数据库：**Microsoft SQL Server 2000【SP3】/2005、Oracle
- **WEBGIS 平台：**MapGIS K9 IMS 平台
- **浏览器：**I IE6 以上系列版本、Firefox 等主流浏览器
- **开发工具：**Microsoft Visual Studio 2005
- **集成开发工具：**Adobe 的 Flex Builder3

## 第2问 怎样配置 MapGIS K9 IMS 开发环境

参照第一章第一问中 MapGIS K9 IMS 的开发环境，在操作系统中安装 .NET 环境、MapGIS K9 IMS 平台、数据库（根据需要选择安装）、浏览器等。

安装 MapGIS K9 IMS 平台后，需要根据开发需求进行 GIS 服务器配置（主要包括 IMS 瓦片数据服务或矢量数据服务）。在二次开发时，必须要启动所要使用的 IMS 数据服务。下面主要介绍平台的安装与 GIS 服务器的配置，详细请查看《MapGIS K9 IMS 平台安装配置篇》文档。

### 1、MapGIS K9 IMS 平台

- （1）先安装证书服务器（若使用网络狗则不需安装）；
- （2）接着安装 MapGIS K9 SP1 二次开发版（MapGIS K9 基础平台）；
- （3）最后安装 MapGIS K9 IMS 增量包。

### 2、数据服务配置

如果要使用瓦片数据服务显示瓦片图，需要打开 MapGIS K9 IMS 服务管理器，在瓦片服务选项卡进行配置。

**服务信息配置：**设置瓦片服务的端口号，一般使用默认值 5142。

**瓦片数据配置：**将二次开发时要用的瓦片数据（裁剪好的 HDF）添加进来，并保存。

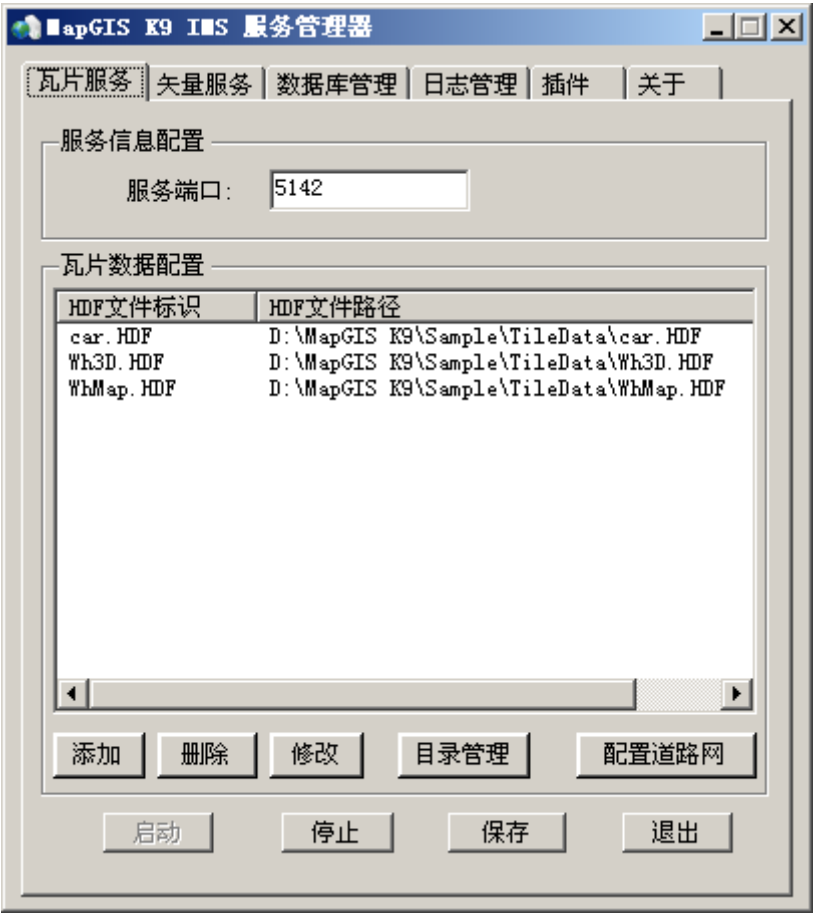


图 2-1 瓦片数据配置

**道路网配置:** 如果要基于瓦片数据服务, 在瓦片地图显示的基础上实现网络分析功能, 则需要对矢量数据库中的道路网文件进行配置。

在平台的 GDB 企业管理器中附加含道路网的矢量地理数据库后, 单击“配置道路网按钮”, 在弹出的对话框中设置道路网的相关信息, 包括道路网名称、数据源、用户名、密码、数据库名称、数据集、网络类。



图 2-2 网络类信息配置

3、数据服务配置

如果要使用矢量数据服务显示矢量的地图文档，并实现基于矢量地图文档的相关 GIS 功能（查询、编辑、分析等），需要在 MapGIS K9 IMS 服务管理器的矢量数据服务选项卡进行配置。

**服务端口号设置：**设置矢量数据服务的端口号，一般是使用默认值 5141。

**地图文档配置（如果基于地图文档发布，需要配置地图文档）：**

（1）地图文档路径：存放地图文档的路径，可设置为 GIS 服务器所在机器硬盘的任一位置。

（2）地图文档发布：单击“发布管理”按钮，打开地图文档发布与管理对话框，将要发布的地图文档添加到相应目录下，设置完成后返回主窗口并保存。

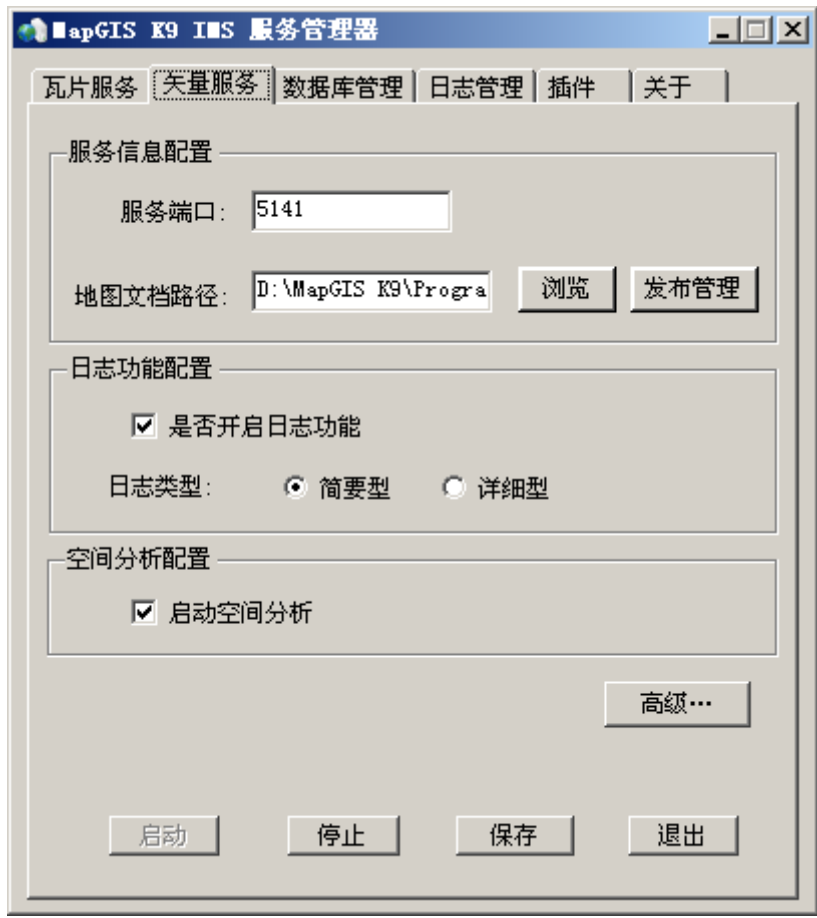


图 2-3 矢量数据配置

在地图文档发布与管理对话框中，可对已添加的地图文档的配置进行编辑修改、删除等操作；也可将存放地图文档的整个目录导入进来；并兼容 MapGIS7.3-IMS 版本配置，可单击“导入旧版配置”按钮进行旧版配置文件的导入。



图 2-4 地图文档发布与管理

**空间分析配置：**若要实现 GIS 的空间分析功能，需要将空间分析配置下的选项勾上，然后打开 MapGIS K9 IMS 服务管理器的数据库管理选项卡，进行空间分析数据库配置，包括数据库类型、数据源、用户名、密码、数据库。（平台提供的空间分析临时库是 SQL 的，支持 Oracle 等主流数据库，配置前要先将数据库附加到相应的数据库管理器中）

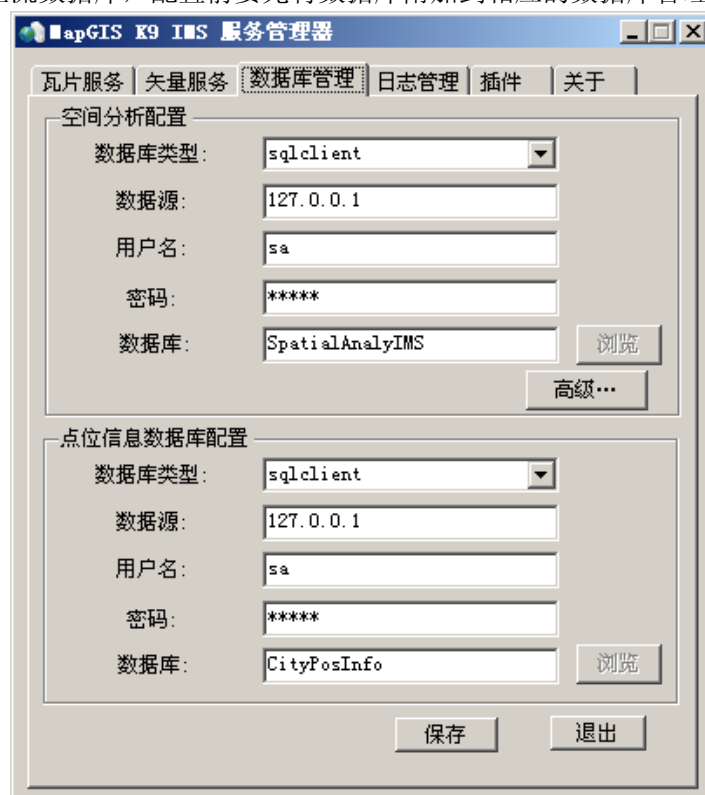


图 2-5 空间分析数据库设置



## 第3问 怎样配置 Flex 开发环境

基于 Flex 的二次开发环境，需要安装配置 Flex Builder 3 集成开发环境，以及两个 FlashPlayer 插件，具体的环境配置如下：

1、Flex Builder 3 集成开发环境，程序安装过程中会自动安装 Flex sdk3；

Adobe 公司 Flex Builder 官方下载地址为：

[http://download.macromedia.com/pub/flex/flex\\_builder/FB3\\_win.exe](http://download.macromedia.com/pub/flex/flex_builder/FB3_win.exe)

Flex sdk3 的下载地址为：

<http://opensource.adobe.com/wiki/display/flexsdk/download?build=3.0.1.2012&pkgtype=1>

2、安装 Flash Palyer 插件，针对不同的浏览器内核选择相应的安装文件进行安装；

**IE 内核**的 flash 播放插件：Adobe Flash Player Active 10.0.22.87.exe

**非 IE 内核**的 flash 播放插件：Adobe Flash Player Plugin 10.0.22.87.exe

Adobe 官方下载地址：

<http://get.adobe.com/flashplayer/>

3、安装 flashplayer 的 debug 程序，用于调试 AS 脚本程序。

**IE 内核**的调试程序：flashplayer\_10\_ax\_debug

**非 IE 内核**的调试程序：

可以到如下网址下载最新安装程序：

[http://fpdownload.macromedia.com/get/flashplayer/current/install\\_flash\\_player\\_ax.exe](http://fpdownload.macromedia.com/get/flashplayer/current/install_flash_player_ax.exe)

## 第4问 怎样创建一个 MapGIS K9 IMS Flex 项目

在 Flex 集成开发环境（Adobe→Adobe Flex Builder 3）中创建 MapGIS K9 IMS Flex 的新工程，具体步骤如下：

1、从开始菜单中打开 Adobe Flex Builder 3，进入 Adobe Flex Builder 3 主界面，选择“NEW”菜单，创建一个 Flex Project 工程，如下图所示：

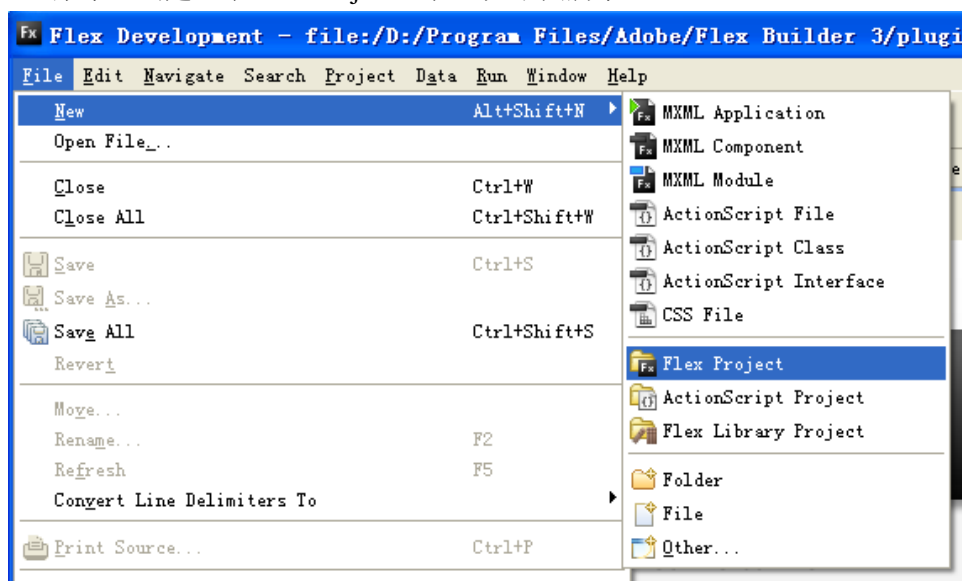


图 4-1 Flex Builder 3 中新建工程

又或者在“Flex Navigator”中，右键空白处，选择“New”→“Flex Project”，如下图所示：

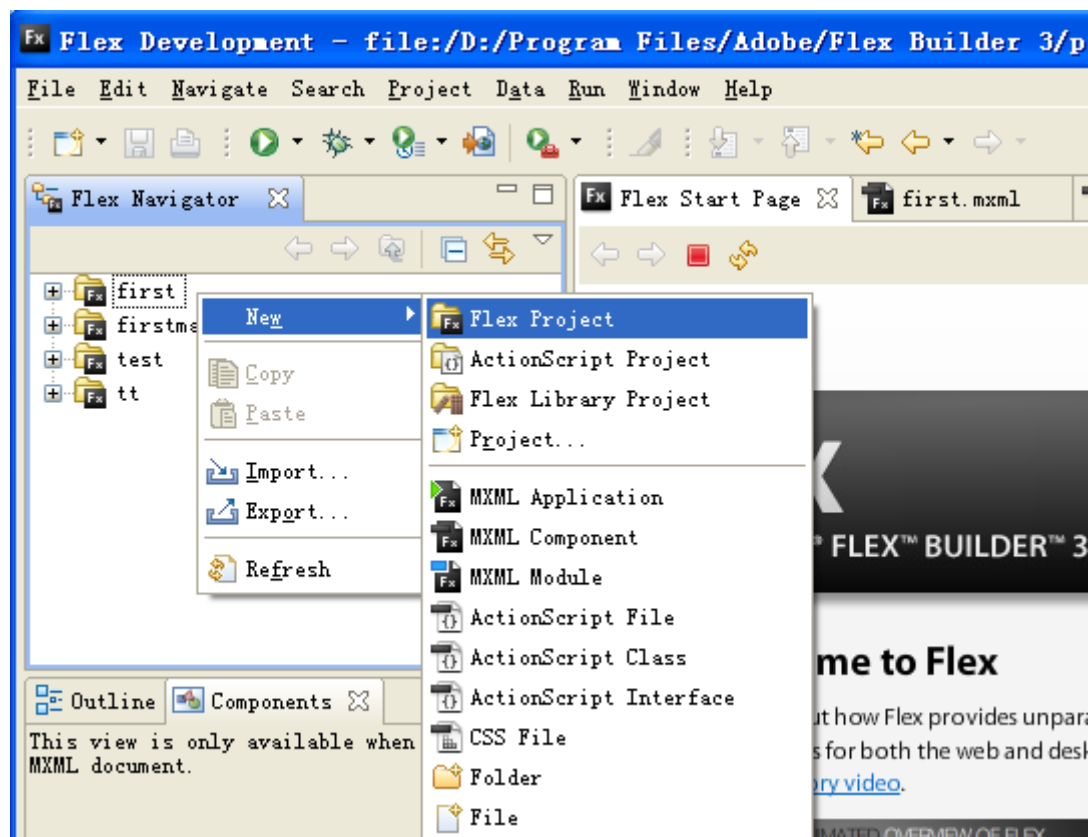


图 4-2 Flex Builder 3 中新建工程（二）

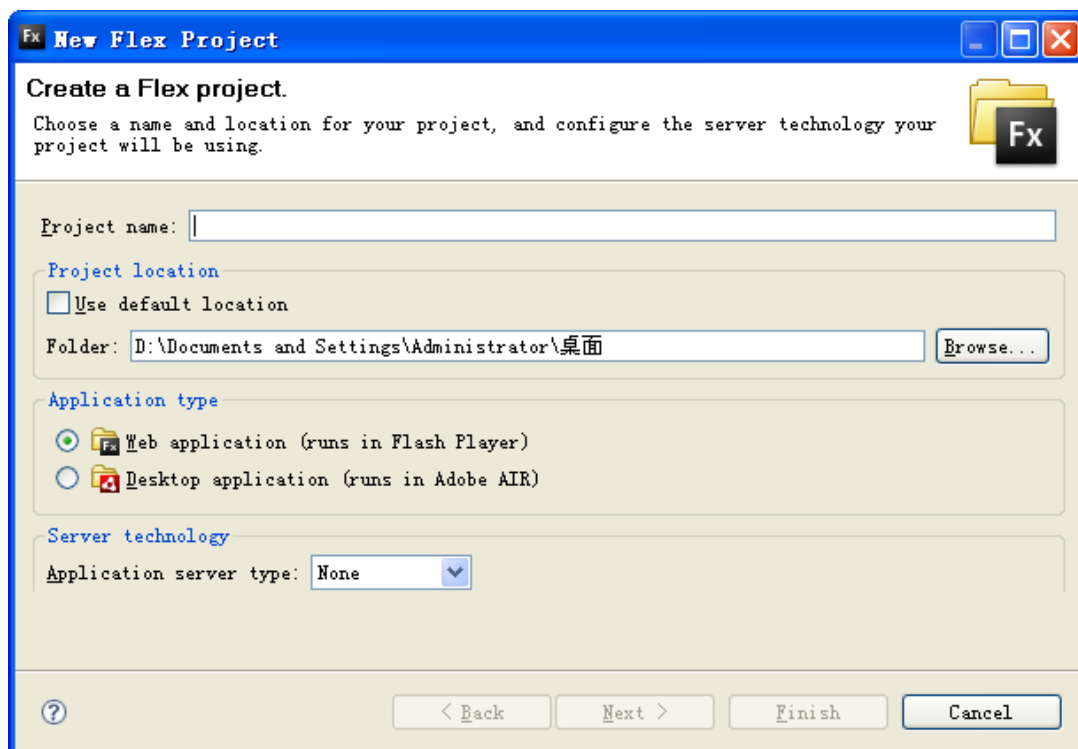


图 4-3 Flex Builder 3 中输入工程名

2、在弹出的对话框中输入工程名，如 demo，选择“Browse”选择工程存放的路径，点击“Next”按钮，如下图所示：

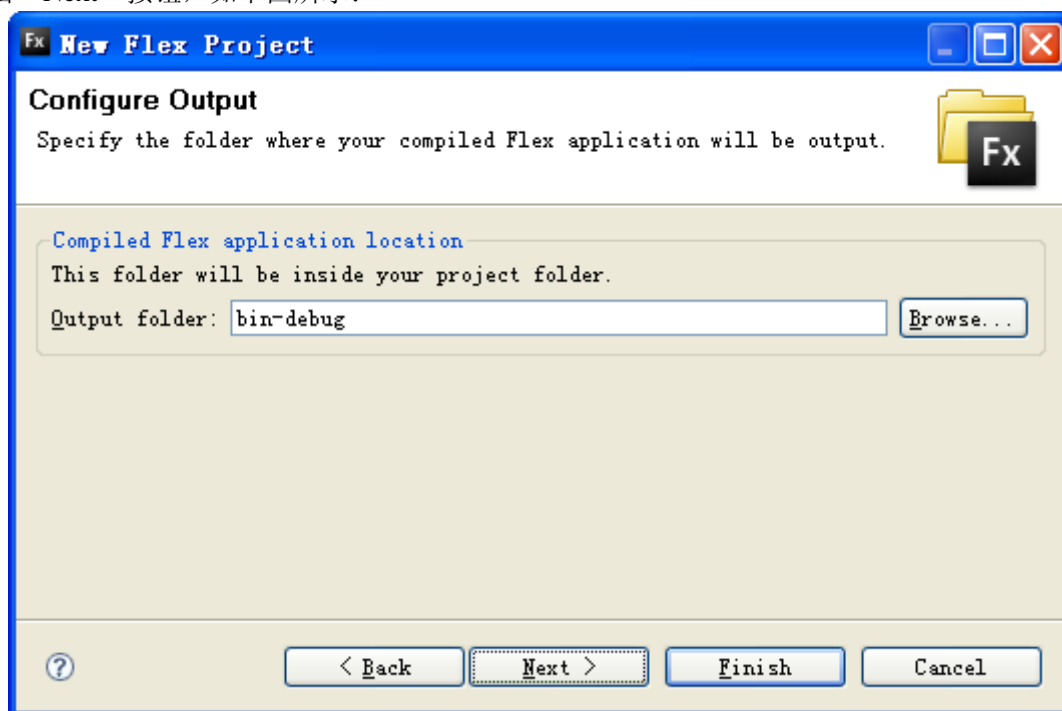


图 4-4 选择工程路径

3、点击“Browse”选择路径，再点击“Next”按钮，进入下一个页面，如图所示：

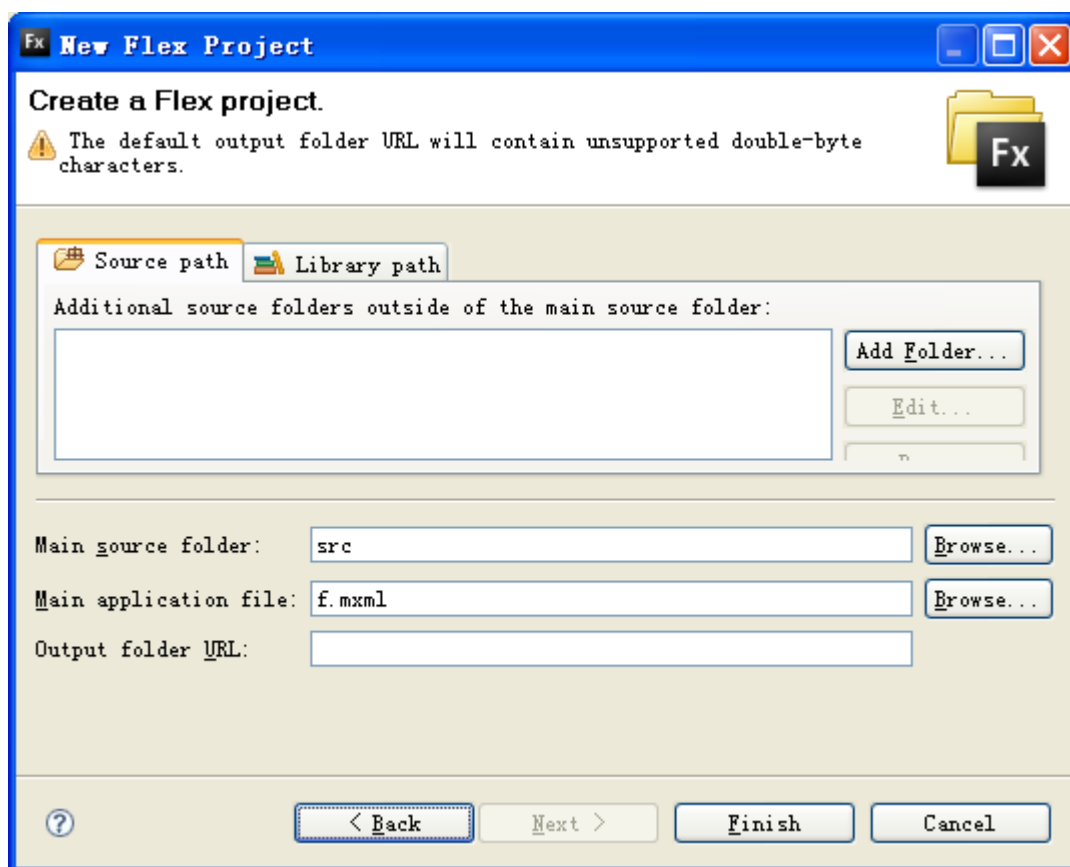


图 4-5 选择工程路径（二）

4、选择“Library path”属性页，选择右边的“SWC”按钮，如下图所示：

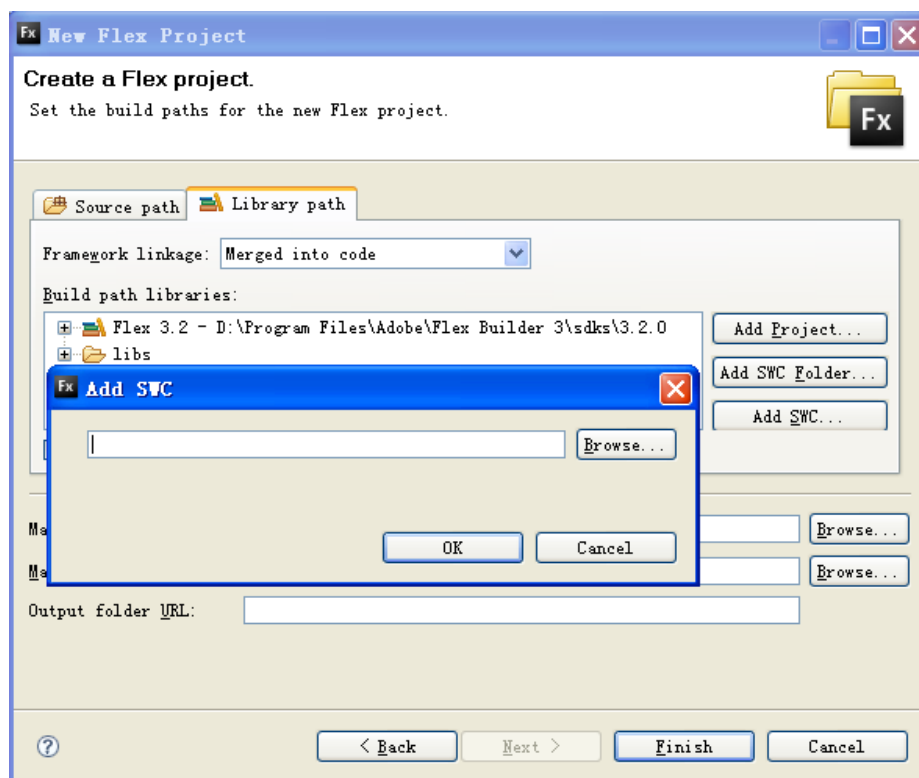


图 4-6 添加 SWC 文件

5、选择 `zdims_2_0_20100720.swc` 文件，该文件位于平台安装目录 `MapGIS K9\WebSDK\Client\Flex\lib` 文件夹中；

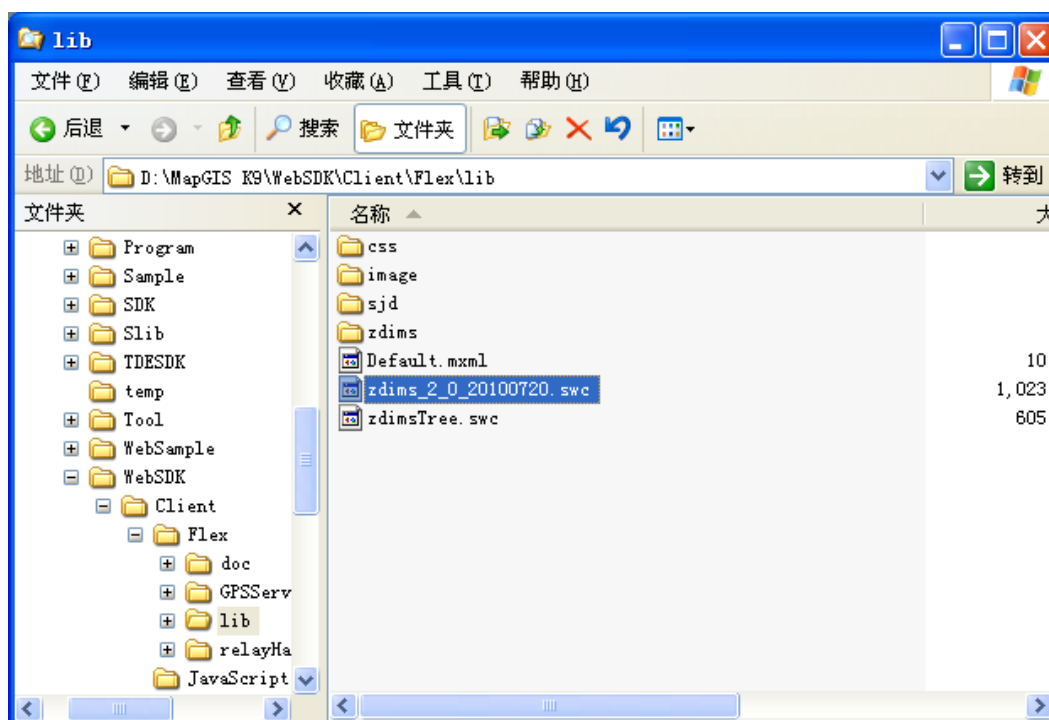


图 4-7 选择 MapGIS K9 IMS 提供的 SWC 文件

6、依次点击“打开”和“OK”按钮，进入如下页面：

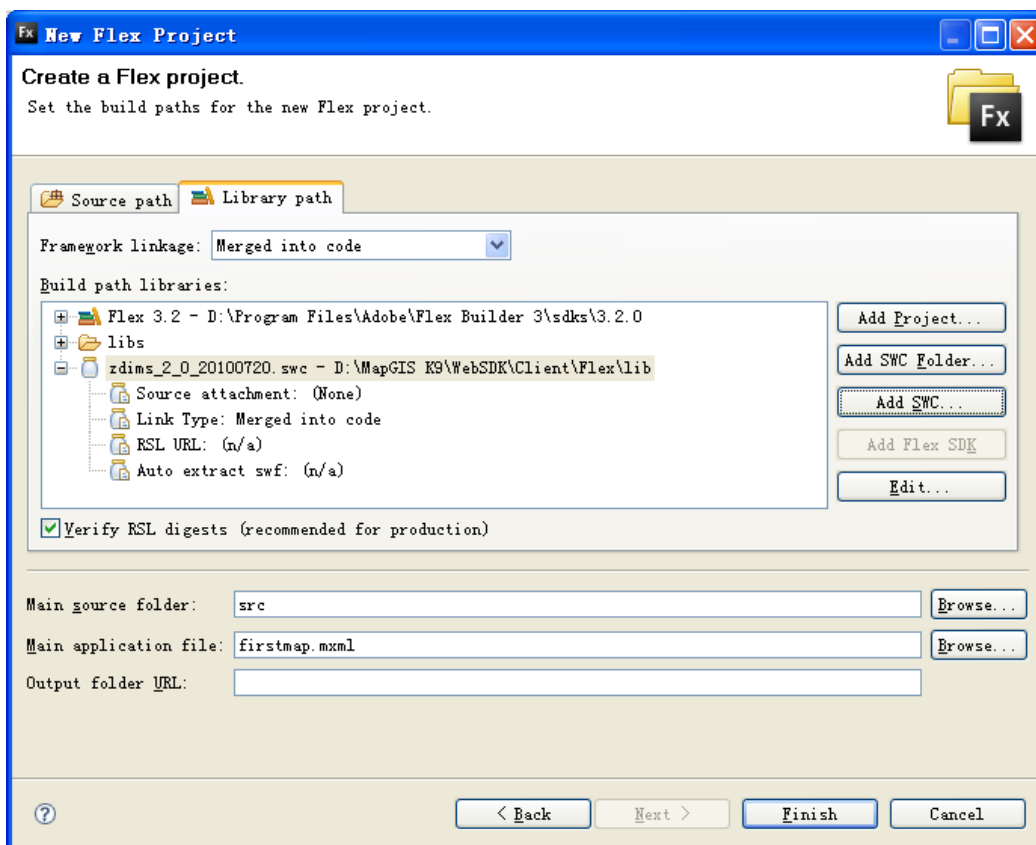


图 4-8 SWC 文件内容

7、点击“Finish”按钮，完成新建工程过程，如下图所示：

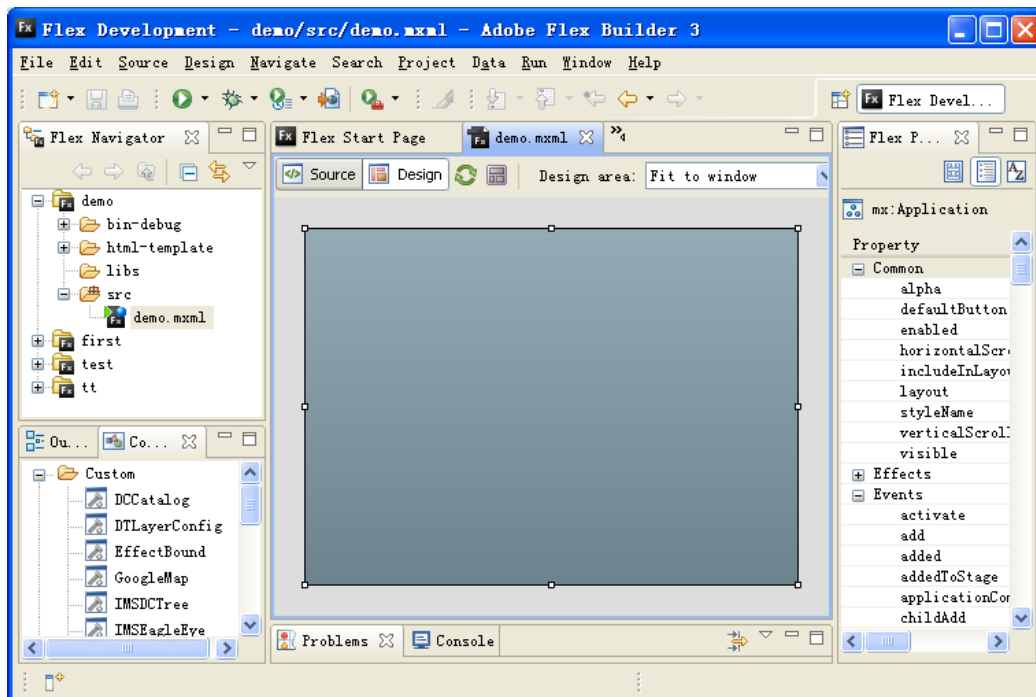


图 4-9 新建工程的主界面

8、工程生成完成后，切换到左边的 Flex Navigator 视图下的 components 窗口中可以看到 custom 目录下面添加了一些自定义控件，如 IMSMAP 等。如下图所示：

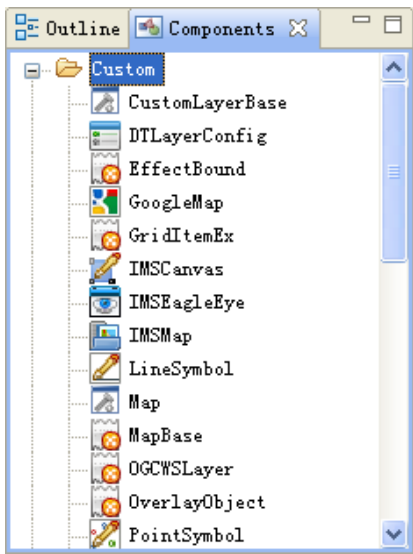


图 4-10 自定义控件

到此为止，一个 Flex project 工程创建完成。现在就可以在此基础上完成基于 MapGIS K9 IMS 平台的二次开发了。

## 第5问 怎样使用 MapGIS K9 IMS Flex 功能组件

MapGIS K9 IMS Flex 功能组件，包括基本功能组件和开源功能组件。

### 1、基础功能组件的使用

MapGIS K9 IMS Flex 的基础功能组件是一个封装好的 SWC，位于 MapGIS K9 IMS 安装目录下，即在平台安装目录 MapGIS K9\WebSDK\Client\Flex\lib 文件夹中。

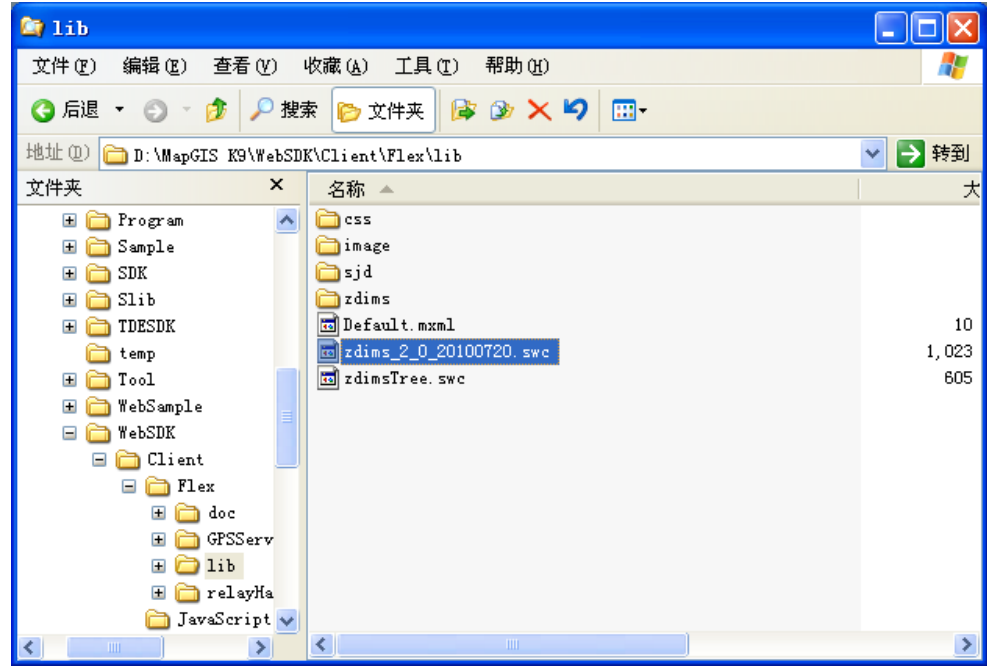


图 5-1 基础功能组件（SWC）

要使用 MapGIS K9 IMS Flex 的基础功能组件，在创建 Flex 项目时直接引用 zdims 的 SWC 库。详细操作可参考第一章第四问创建一个 MapGIS K9 IMS Flex 工程的操作。

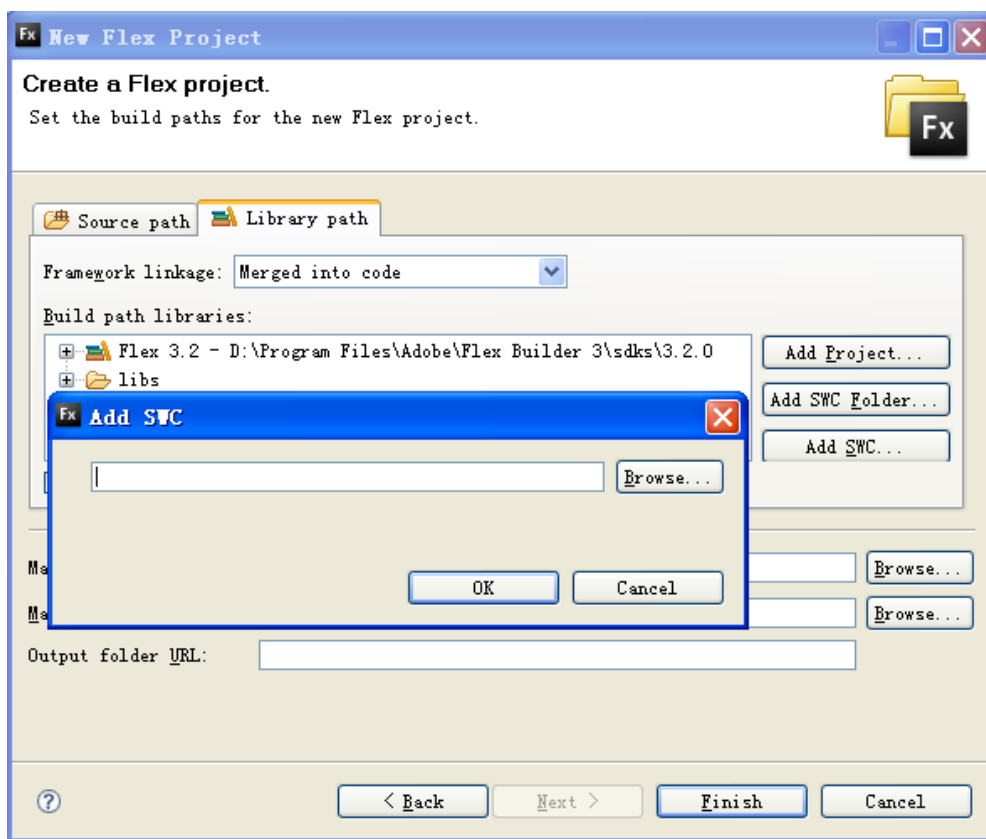


图 5-2 引用 zdims 的 swc (一)

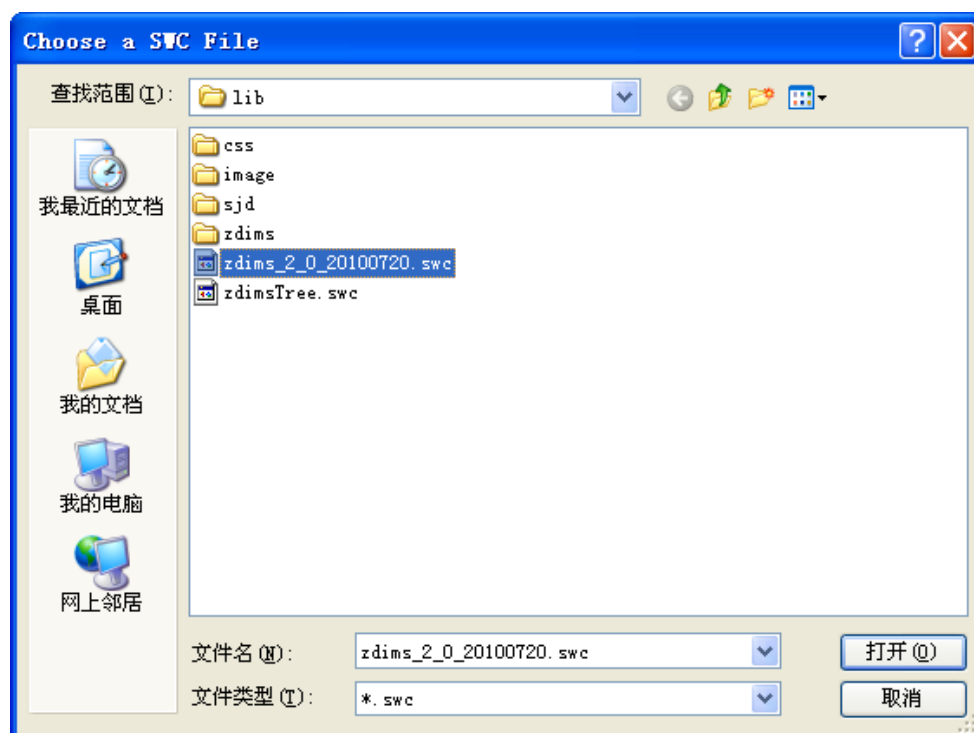


图 5-3 引用 zdims 的 swc (二)

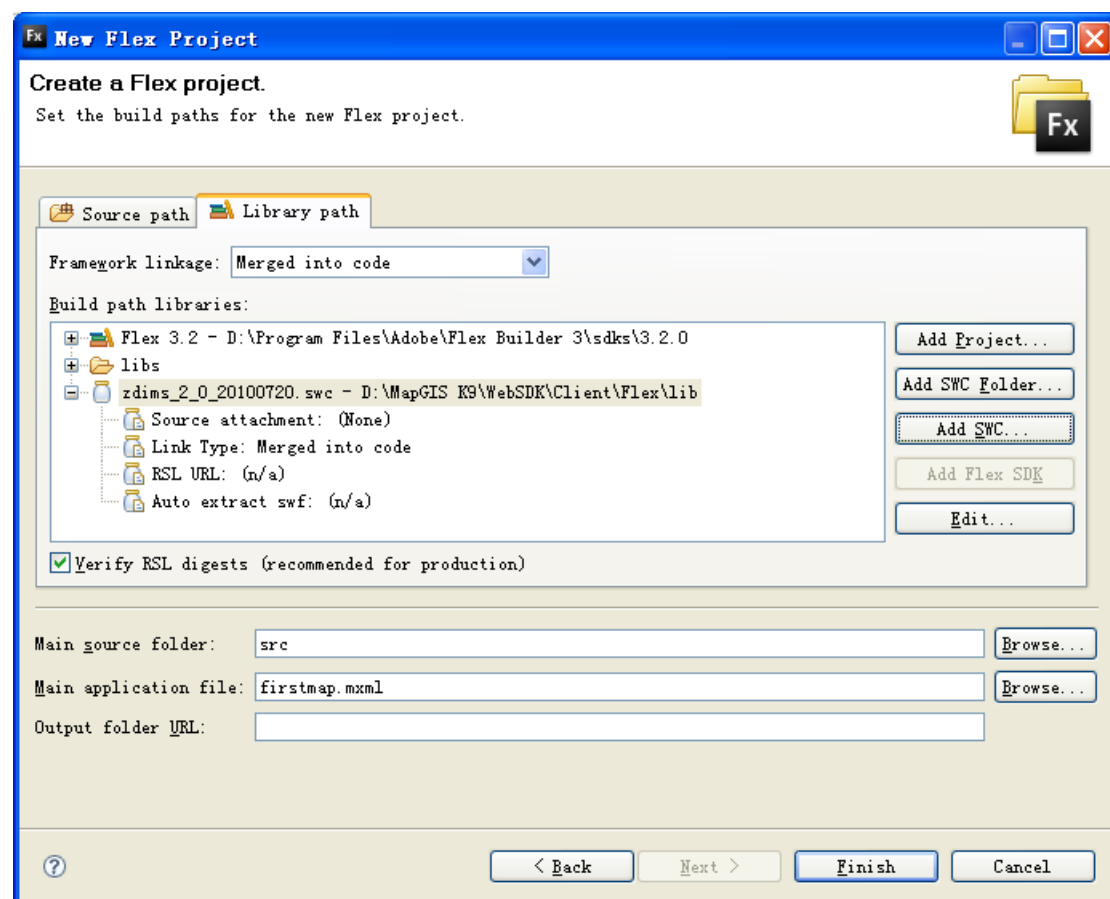


图 5-4 引用 zdims 的 swc (三)

## 2、开源功能组件的使用

MapGIS K9 IMS Flex 的开源功能组件，位于 MapGIS K9 IMS 安装目录下，即在平台安装目录 MapGIS K9\WebSDK\Client\Flex\lib 中。

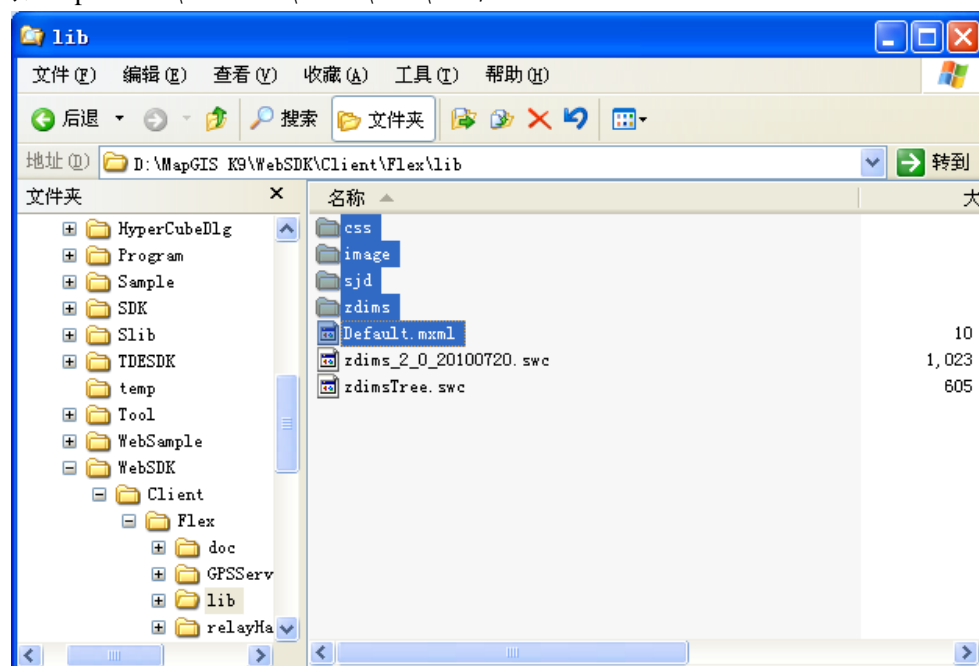


图 5-5 开源功能组件



要使用 MapGIS K9 IMS Flex 的开源功能组件,直接将 lib 下的相关文件拷贝到所用工程的 src 目录下,在左侧的 Flex Navigator 视图 components 中的 custom 目录中自动显示全部功能控件。在开发中,直接拖动相应功能控件即可。

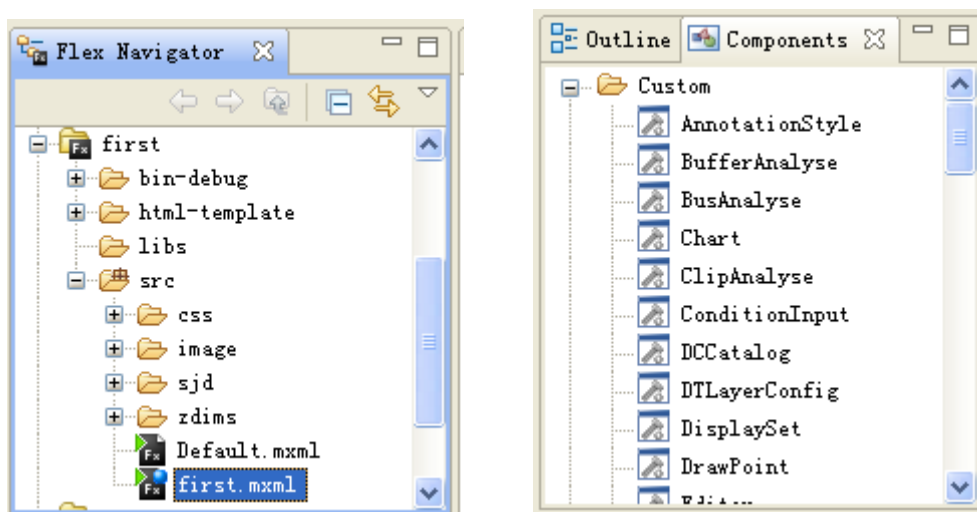


图 5-6 开源功能组件列表

## 第6问 怎样更新 MapGIS K9 IMS Flex 功能组件

要更新 MapGIS K9 IMS Flex 功能组件,分别将基本功能组件库和开源功能组件库替换。具体操作方法如下:

### 1、更新基础功能组件

(1) 在 Adobe Flex Builder 3 集成开发环境中,选中要更新的工程,右键单击该工程,选择“Properties”,打开该工程的属性设置对话框。

(2) 在工程的属性设置对话框中,在左侧目录中选择“Flex Build Path”,切换到“Library Path”,选中原来的基础功能库 (swc),单击右侧的“Remove”按钮,将基础库移除。

(3) 移除原基础库后,单击右侧的“Add SWC”按钮,选择要更新的基础功能库 (swc) 后,单击“OK”按钮,将新的 SWC 添加到工程中,更新完毕。

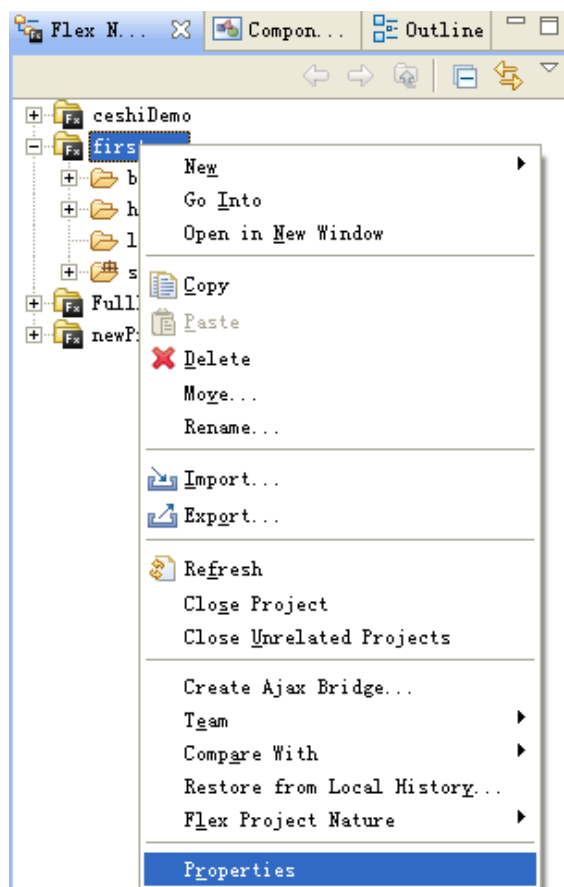


图 6-1 右键打开工程的属性

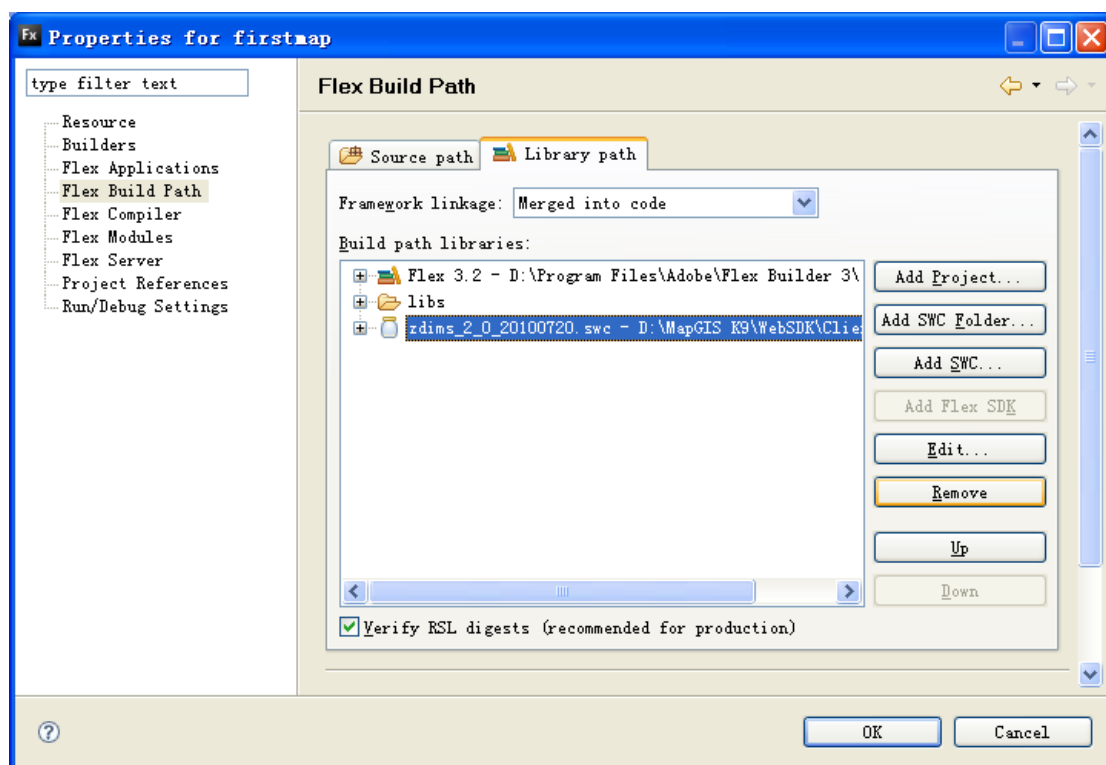


图 6-2 移除出来的基础功能组件

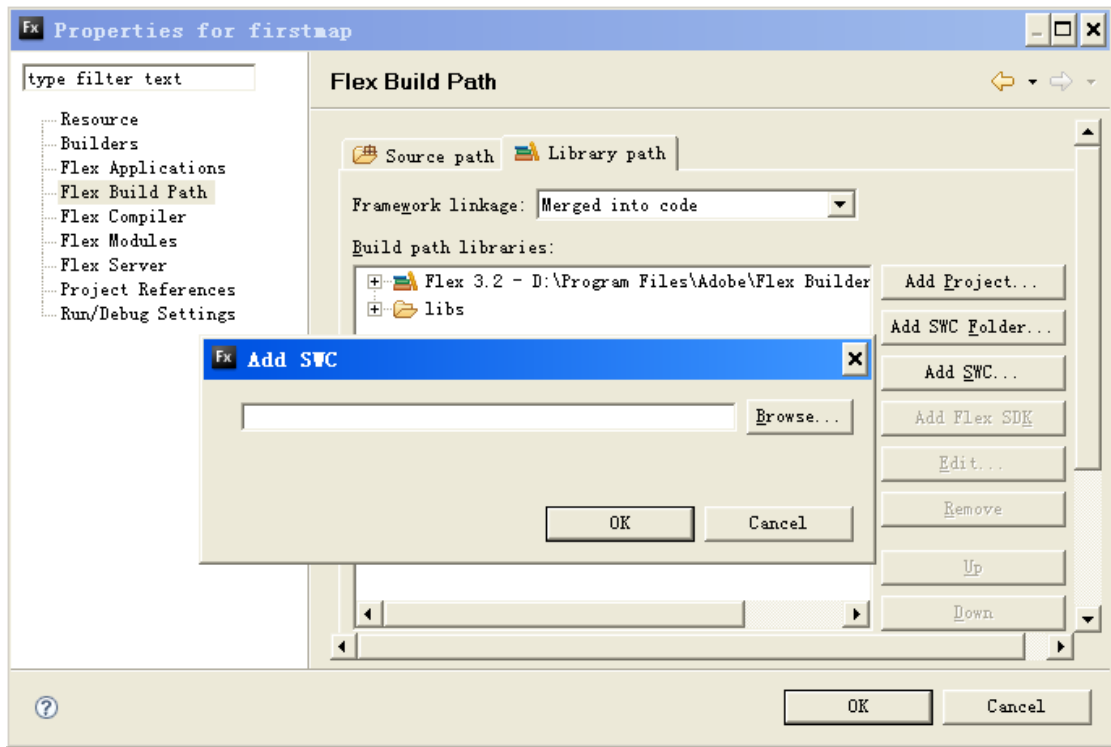


图 6-3 添加新的基础功能组件

## 2、更新开源功能组件

更新 MapGIS K9 IMS Flex 的开源功能组件，直接将新的开源功能库文件（如 MapGIS K9\WebSDK\Client\Flex\lib 下的相关文件）拷贝到工程的 src 目录下，在弹出的对话框中选择“**Yes to All**”，将原来的功能库全部覆盖。

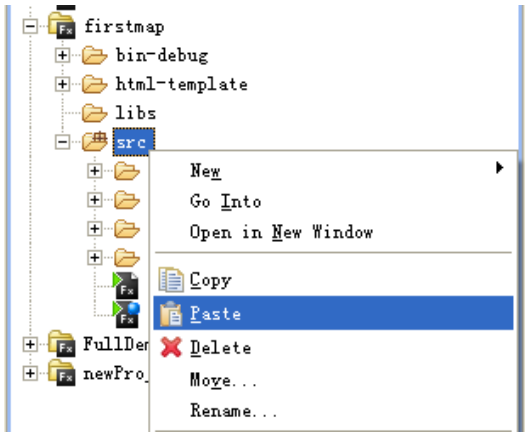


图 6-4 拷贝新的开源功能库

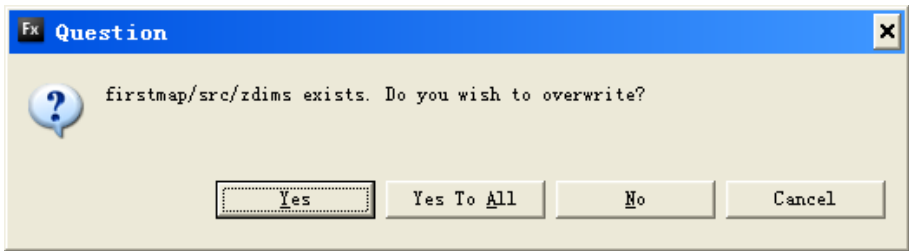


图 6-5 全部覆盖原来的功能库

## 第二章 地图显示篇

### 第1问 怎样使用地图容器

#### 控件式:

新建一个 MXML Application,切换到设计界面,在 Components 选项卡下的 Custom 目录下,找到 IMSMap 控件,将其拖放至设计界面。切换到源码显示窗口,显示如下:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMap x="34" y="22" width="703" height="594">
    </ns1:IMSMap>
</mx:Application>
```

#### 代码式:

通过代码创建地图容器非常简单,只需在需要显示地图的时候创建一个地图容器类,然后将其作为子节点添加到 MXML Application 中即可。下面的代码实现了在 MXML Application 界面创建完成时用代码动态的创建一个地图容器。

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" creationComplete="addIMSMap()">
<mx:Script>
    <![CDATA[
        import zdims.map.IMSMap;
        public function addIMSMap():void
        {
            var map:IMSMap=new IMSMap();
            map.width=500;
            map.height=500;
            this.addChild(map);
        }
    ]]>
</mx:Script>
</mx:Application>
```

## 第2问 怎样显示瓦片地图

### 控件式:

参照第二章第 1 问创建一个地图容器，给地图容器的 levelNum 属性和显示范围属性赋值。然后在 Components 选项卡下的 Custom 目录下，找到 TileLayer 控件，将其拖放至地图容器 IMSMap 中，然后打开 Properties 窗口，切换到 Category View，展开 MapGisIMS 节点，修改瓦片数据名称 hdfName 和服务地址 serverAddress 属性。切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMap x="34"
        y="22"
        width="703"
        height="594"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMap>
</mx:Application>
```

### 代码式:

参照第二章第 1 问创建一个地图容器，给地图容器的总显示级数 levelNum 和显示范围赋值，然后创建一个瓦片图层类，给瓦片数据名称 hdfName 和服务地址 serverAddress 属性赋值，并将其作为子节点添加到地图容器对象中。下面的代码实现了在 MXML Application 界面创建完成时动态添加一个瓦片地图。

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" creationComplete="addIMSMap()">
<mx:Script>
    <![CDATA[
        import zdims.map.TileLayer;
        import zdims.map.IMSMap;
        public function addIMSMap():void
```

```

    {
        var map:IMSMMap=new IMSMap();
        map.width=500;
        map.height=500;
        map.levelNum=7;
        map.xMinMap=114.125602229914;
        map.xMaxMap=114.500788705197;
        map.yMinMap=30.4539323507469;
        map.yMaxMap=30.8291188260302;
        var tileLayer:TileLayer=new TileLayer();
        tileLayer.hdfName="whmap.hdf";
        tileLayer.serverAddress="127.0.0.1:5142";
        map.addChild(tileLayer);
        this.addChild(map);
    }
]]>
</mx:Script>
</mx:Application>

```

### 第3问 怎样显示地图文档

#### 控件式:

参照第二章第 1 问创建一个地图容器，给地图容器的显示范围属性赋值。然后在 Components 选项卡下的 Custom 目录下，找到 VectorMapDoc 控件，将其拖放至地图容器 IMSMap 中，然后打开 Properties 窗口，切换到 Category View，展开 MapGisIMS 节点，修改地图文档名称 mapDocName 和服务器地址 serverAddress 属性。切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMMap x="34"
        y="22"
        width="703"
        height="594"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMMap>
</mx:Application>

```

```

    </ns1:IMSMAP>
</mx:Application>

```

## 代码式:

参照第二章第 1 问创建一个地图容器，给地图容器的显示范围赋值，然后创建一个地图文档类，给地图文档名称 mapDocName 和服务地址 serverAddress 属性赋值，并将其作为子节点添加到地图容器对象中。下面的代码实现了在 MXML Application 界面创建完成时用代码动态添加一个地图文档。

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" creationComplete="addIMSMAP()">
<mx:Script>
    <![CDATA[
        import zdims.map.VectorMapDoc;
        import zdims.map.IMSMAP;
        public function addIMSMAP():void
        {
            var map:IMSMAP=new IMSMAP();
            map.width=500;
            map.height=500;
            map.xMinMap=114.125602229914;
            map.xMaxMap=114.500788705197;
            map.yMinMap=30.4539323507469;
            map.yMaxMap=30.8291188260302;
            var mapdoc:VectorMapDoc=new VectorMapDoc();
            mapdoc.mapDocName="wh.map";
            mapdoc.serverAddress="127.0.0.1:5141";
            map.addChild(mapdoc);
            this.addChild(map);
        }
    ]]>
</mx:Script>
</mx:Application>

```

## 第4问 怎样叠加显示瓦片地图和地图文档

### 控件式:

参照第二章第 2 问创建一个地图容器并显示一个瓦片地图控件，再参照第二章第 3 问添

加一个地图文档控件到 **IMSMap**。切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMap x="34"
        y="22"
        width="703"
        height="594"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
        <ns1:VectorMapDoc mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMap>
</mx:Application>
```

## 代码式：

参照第二章第 2 问创建一个地图容器并添加一个瓦片图层对象，然后参照第二章第 3 问添加一个地图文档对象。下面的代码实现了在 **MXML Application** 界面创建完成时用代码动态添加一个瓦片地图和地图文档进行叠加显示。

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" creationComplete="addIMSMap()">
<mx:Script>
    <![CDATA[
        import zdims.map.VectorMapDoc;
        import zdims.map.TileLayer;
        import zdims.map.IMSMap;
        public function addIMSMap():void
        {
            var map:IMSMap=new IMSMap();
            map.width=500;
            map.height=500;
            map.levelNum=7;
            map.xMinMap=114.125602229914;
            map.xMaxMap=114.500788705197;
```



```

        map.yMinMap=30.4539323507469;
        map.yMaxMap=30.8291188260302;
        var tileLayer:TileLayer=new TileLayer();
        tileLayer.hdfName="whmap.hdf";
        tileLayer.serverAddress="127.0.0.1:5142";
        map.addChild(tileLayer);
        var mapdoc:VectorMapDoc=new VectorMapDoc();
        mapdoc.mapDocName="wh.map";
        mapdoc.serverAddress="127.0.0.1:5141";
        map.addChild(mapdoc);
        this.addChild(map);
    }
    ]]>
</mx:Script>
</mx:Application>

```

## 第5问 怎样同时显示多个地图窗口

### 控件式:

以分别在容器左半部分和右半部分分别显示一个地图窗口为例, 首先参照第二章第 2 问创建一个地图容器并显示一个瓦片地图控件, 重复操作两次, 调整两个 IMSMap 的位置, 分别位于窗口左面和右面。切换到源码显示窗口, 显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMap x="10"
        y="22"
        width="449"
        height="544"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMap>
    <ns1:IMSMap x="484"
        y="22"
        width="466"

```

```

        height="544"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
<ns1:TileLayer hdfName="whmap.hdf"
        serverAddress="127.0.0.1:5142">
</ns1:TileLayer>
</ns1:IMSMAP>
</mx:Application>

```

## 代码式:

参照第二章第2问创建一个地图容器类并添加一个瓦片图层对象,重复两次,分别在窗口左面和右面添加一个显示瓦片的地图容器。下面的代码实现了在 MXML Application 界面创建完成时用代码动态添加两个地图容器,并在窗口左右两部分分别显示。

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" creationComplete="addIMSMAP()">
<mx:Script>
    <![CDATA[
        import zdims.map.TileLayer;
        import zdims.map.IMSMAP;
        public function addIMSMAP():void
        {
            var map:IMSMAP=new IMSMAP();
            map.width=500;
            map.height=500;
            map.levelNum=7;
            map.xMinMap=114.125602229914;
            map.xMaxMap=114.500788705197;
            map.yMinMap=30.4539323507469;
            map.yMaxMap=30.8291188260302;
            var tileLayer:TileLayer=new TileLayer();
            tileLayer.hdfName="whmap.hdf";
            tileLayer.serverAddress="127.0.0.1:5142";
            map.addChild(tileLayer);
            var map_right:IMSMAP=new IMSMAP();
            map_right.x=500;
            map_right.width=500;
            map_right.height=500;
            map_right.levelNum=7;

```

```

        map_right.xMinMap=114.125602229914;
        map_right.xMaxMap=114.500788705197;
        map_right.yMinMap=30.4539323507469;
        map_right.yMaxMap=30.8291188260302;
        var tileLayer_r:TileLayer=new TileLayer();
        tileLayer_r.hdfName="whmap.hdf";
        tileLayer_r.serverAddress="127.0.0.1:5142";
        map_right.addChild(tileLayer_r);
        this.addChild(map);
        this.addChild(map_right);
    }
}]>
</mx:Script>
</mx:Application>

```

## 第6问 怎样添加一张图片跟地图叠加显示

参照第二章第2问创建一个地图容器并显示一个瓦片地图控件,再参照第二章第3问添加一个地图文档控件到 **IMSMa**p。切换到源码显示窗口,显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*" creationComplete="addImg()">
    <ns1:IMSMa p id="imsma p"
        x="34"
        y="22"
        width="703"
        height="594"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMa p>
</mx:Application>
<mx:Script>
    <![CDATA[
        import zdims.control.Marker;
        public function addImg():void
        {
            var mark:Marker=new Marker();
            mark.logicX=114.3;

```

```

        mark.logicY=30.6;
        mark.setIconSrc("image/mark/walk.gif");
        this.imsmap.addChild(mark);
    }
]]>
</mx:Script>
</mx:Application>

```

## 第7问 怎样去除标注弹出窗口的图片

如果添加标注的时候给标注控件的 `markerInfo` 属性赋了值, 则鼠标点击标注的时候会弹出标注信息框。如果默认的信息框不能满足实际项目需要, 可以对其进行修改。在 `zdims\control` 文件夹下找到 `MarkerContent.mxml`, 双击打开, 切换到设计界面, 然后就可以对其进行修改, 例如: 要去除信息窗口中的图片控件, 在设计界面将其删除即可。

## 第8问 怎样使地图容器与鹰眼显示不同地图

### 控件式:

鹰眼显示的地图跟地图窗口类似只需要将需要显示的图层控件拖放至鹰眼控件中即可, 下面显示了以地图容器显示瓦片地图 `whmap.hdf`, 鹰眼显示地图文档 `wh.map` 为例的效果。切换到源码显示窗口, 显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMap x="46"
        y="43"
        width="584"
        height="400"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302"
        layout="absolute">
        <ns1:IMSEagleEye x="373"
            y="249"
            width="211"
            height="151"
            eagleLocation="右下">
            <ns1:VectorMapDoc mapDocName="wh.map">

```

```

        </ns1:VectorMapDoc>
    </ns1:IMSEagleEye>
    <ns1:TileLayer x="77"
        y="56"
        hdfName="whmap.hdf">
    </ns1:TileLayer>
</ns1:IMSMap>
</mx:Application>

```

## 代码式:

参照第二章第 2 问创建一个地图容器类并添加一个瓦片图层对象,然后在创建一个鹰眼对象,再创建一个地图文档对象并将其添加到鹰眼对象中。最后把鹰眼添加到地图容器中。下面的代码实现了在 **MXML Application** 界面创建完成时用代码动态添加一个地图容器和鹰眼,并显示不同的地图类型。

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute" creationComplete="addIMSMap()">
<mx:Script>
    <![CDATA[
        import zdims.util.EagleLocation;
        import zdims.map.VectorMapDoc;
        import zdims.map.IMSEagleEye;
        import zdims.map.TileLayer;
        import zdims.map.IMSMap;
        public function addIMSMap():void
        {
            var map:IMSMap=new IMSMap();
            map.width=500;
            map.height=500;
            map.levelNum=7;
            map.xMinMap=114.125602229914;
            map.xMaxMap=114.500788705197;
            map.yMinMap=30.4539323507469;
            map.yMaxMap=30.8291188260302;
            var tileLayer:TileLayer=new TileLayer();
            tileLayer.hdfName="whmap.hdf";
            tileLayer.serverAddress="127.0.0.1:5142";
            map.addChild(tileLayer);
            this.addChild(map);
            var eagle:IMSEagleEye=new IMSEagleEye();
            eagle.width=200;
            eagle.height=200;

```

```

        eagle.eagleLocation=EagleLocation.BottomRight;
        var doc:VectorMapDoc=new VectorMapDoc();
        doc.mapDocName="wh.map";
        eagle.addChild(doc);
        map.addChild(eagle);
    }
]]>
</mx:Script>
</mx:Application>

```

## 第9问 怎样切换鹰眼显示和隐藏状态

参照第二章第8问添加一个鹰眼，然后再添加一个Button控件，给其click事件赋值，在这里我写了一个Button按钮点击事件处理函数switchEagle，当鼠标点击该按钮的时候，该函数调用鹰眼对象的switchEagleEye函数切换鹰眼控件的显示和隐藏状态。切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSTMap x="46"
        y="43"
        width="584"
        height="400"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302"
        layout="absolute">
        <ns1:IMSEagleEye id="eagle" x="373"
            y="249"
            width="211"
            height="151"
            eagleLocation="右下">
            <ns1:VectorMapDoc mapDocName="wh.map">
            </ns1:VectorMapDoc>
        </ns1:IMSEagleEye>
        <ns1:TileLayer x="77"
            y="56"
            hdfName="whmap.hdf">
        </ns1:TileLayer>
    </ns1:IMSTMap>
    <mx:Button x="100" y="10" label="显示/隐藏鹰眼" click="switchEagle()"

```

```

fontSize="12"/>
    <mx:Script>
        <![CDATA[
            public function switchEagle():void
            {
                this.eagle.switchEagleEye();
            }
        ]]>
    </mx:Script>
</mx:Application>

```

## 第10问 怎样实现不同的客户端看到的地图不同

要实现不同的客户端显示不同的地图有两种解决方案：

1. 服务器端主动实现：在 Flash 应用程序中，主动识别客户端用户，如采用用户登录方式。在应用程序中可以根据登录的用户类型动态的修改地图容器中的地图。
2. 客户端主动实现：在 Flash 应用程序中可以把地图数据列表展现出来，让客户端用户自由选择显示什么地图数据。

下面以客户端主动实现为例，实现瓦片地图和地图文档地图数据的自由切换。在该示例中我添加了两个 Button 控件，点击第一个调用 showHDF 函数显示瓦片地图，点击第二个调用 showDoc 显示地图文档。

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMap id="imsmap" x="46"
        y="43"
        width="584"
        height="400"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302"
        layout="absolute">
    </ns1:IMSMap>
    <mx:Script>
        <![CDATA[
            import zdims.map.TileLayer;
            import zdims.map.VectorMapDoc;
            private var doc:VectorMapDoc=null;
            private var tileLayer:TileLayer=null;
            public function showHDF():void
            {

```

```

        if (doc != null)
            this.imsmap.removeChild(doc);
        if (tileLayer == null)
        {
            tileLayer = new TileLayer();
            tileLayer.hdfName = "whmap.hdf";
        }
        this.imsmap.addChild(tileLayer);
    }
    public function showDoc():void
    {
        if (tileLayer != null)
            this.imsmap.removeChild(tileLayer);
        if (doc == null)
        {
            doc = new VectorMapDoc();
            doc.mapDocName = "wh.map";
        }
        this.imsmap.addChild(doc);
    }
    ]]>
</mx:Script>
<mx:Button x="82"
           y="13"
           label="点我显示瓦片whmap.hdf"
           click="showHDF()"
           fontSize="12"/>
<mx:Button x="296"
           y="13"
           label="点我显示地图文档wh.map"
           click="showDoc()"
           fontSize="12"/>
</mx:Application>

```

## 第11问 怎样实现鼠标点击或移到某个对象上获取某对象的属性

该问题的关键在于要捕捉对象的鼠标点击事件和鼠标移过事件。参照第二章第2问实现瓦片地图显示，然后拖动两个 Button 控件到界面中，给第一个 Button 的鼠标点击事件赋值响应 getWidth 函数，给第二个 Button 的鼠标移过事件赋值响应 getHeight 函数。下面的示例实现了鼠标点击或移过按钮的时候获取地图容器的高和宽属性，示例代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
```



```

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMMap id="imsmap"
        x="46"
        y="43"
        width="584"
        height="400"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302"
        layout="absolute">
        <ns1:TileLayer x="77"
            y="56"
            hdfName="whmap.hdf">
        </ns1:TileLayer>
    </ns1:IMSMMap>
    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;
            public function getWidth():void
            {
                Alert.show("width="+this.imsmap.width.toString());
            }
            public function getHeight():void
            {
                Alert.show("height="+this.imsmap.height.toString());
            }
        ]]>
    </mx:Script>
    <mx:Button x="59"
        y="11"
        label="用鼠标点我获取地图容器的宽"
        mouseDown="getWidth()"
        fontSize="12"/>
    <mx:Button x="266"
        y="11"
        label="鼠标靠近我获取地图容器的高"
        mouseOver="getHeight()"
        fontSize="12"/>
</mx:Application>

```

## 第12问 什么原因会引起地图不能正常显示

1、如果地图容器包含瓦片地图，一定要给地图容器的终止显示级数 `levelNum` 和显示范围 (`xMinMap`, `yMinMap`, `xMaxMap`, `yMaxMap`) 赋值，该情况下显示范围必须是瓦片地图的截图范围，否则不能正常显示和使用瓦片地图。

2、如果地图容器里只包含矢量地图，则不需要设置地图容器的终止显示级数 `levelNum`，但显示范围 (`xMinMap`, `yMinMap`, `xMaxMap`, `yMaxMap`) 属性一定要设置，该情况下的显示范围属性可以设置为任何自己想要初始化时显示的地图范围。

## 第13问 什么原因导致本机可以出图但其他机器访问不能出图

1、如果本地访问可以出图，发布 Flex 应用程序后在远程客户端如果不能正常如图，则可能原因是客户端浏览器设置了代理导致不能访问到 Flex 应用程序。

2、如果用 Flex Builder 调试或运行程序能正常显示地图，而发布后不能正常显示地图，则可能是本地计算机设置了代理导致 Flex 应用发布后不能访问到 GIS 服务器。

## 第14问 怎样在网页中嵌入 MapGIS K9 IMS Flex 应用

有时需要将自己做好的 Flex 程序嵌入到网页中的某个位置，如何嵌入呢？其实非常简单，只需要在网页中的某个位置插入一个 `object` 标签即可。下面的示例实现了在网页中插入一个 Flex 应用 `C02_11_01.swf`。示例代码如下：

```
<html>
<body>
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
        id="Default" width="100%" height="100%"
        codebase="http://fpdownload.macromedia.com/get/flashplayer/current/swflash.cab">
    <param name="movie" value="../bin-debug/C02_11_01.swf" />
    <param name="quality" value="high" />
    <param name="bgcolor" value="#869ca7" />
    <param name="allowScriptAccess" value="sameDomain" />
</object>
</body>
</html>
```

## 第三章 地图控制篇

### 第1问 怎样复位地图

首先参照第二章第 2 问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮的时候让地图复位，则需要进行以下几步操作：首先，添加一个 onclick() 函数，用来处理按钮的鼠标点击事件，这个函数调用 IMSMap 控件的 setCurOper 函数，传入参数为 IMSOperType.Restore，表示复位地图操作；然后给 Button 控件的 click 属性赋值为 onclick()。切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMap>
    <mx:Button x="160" y="34" label="复位" click="onclick()"/>
    <mx:Script>
        <![CDATA[
            import zdims.util.IMSOperType;
            public function onclick():void
            {
                this.map.setCurOper(IMSOperType.Restore);
            }
        ]]>
    </mx:Script>
</mx:Application>
```

## 第2问 怎样刷新地图

首先参照第二章第 2 问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮的时候让地图刷新，则需要进行以下几步操作：首先，添加一个 onclick() 函数，用来处理按钮的鼠标点击事件，这个函数调用 IMSMap 控件的 setCurOper 函数，传入参数为 IMSOperType.Refresh，表示刷新地图操作；然后给 Button 控件的 click 属性赋值为 onclick()。切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMap>
    <mx:Button x="160" y="34" label="刷新" click="onclick()" />
    <mx:Script>
        <![CDATA[
            import zdims.util.IMSOperType;
            public function onclick():void
            {
                this.map.setCurOper(IMSOperType.Refresh);
            }
        ]]>
    </mx:Script>
</mx:Application>
```

## 第3问 怎样控制地图显示级别

首先参照第二章第 2 问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮的时候控制地图的显示级别，例如：让地图立即显示第三级，需要进行以下几步操作：首先，添加一个 onclick() 函数，用来处理按钮的鼠标点击事件，这个函数调用 IMSMap 控件的 setLevel 函数，传入参数为 3，表示让地图显示第三级数据；然后给 Button

控件的 click 属性赋值为 onclick()。切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMMap>
    <mx:Button x="160" y="34" label="显示第3级" click="onclick()" />
    <mx:Script>
        <![CDATA[
            public function onclick():void
            {
                this.map.setLevel(3);
            }
        ]]>
    </mx:Script>
</mx:Application>
```

## 第4问 怎样使地图在某一显示级别居中显示某个位置

首先参照第二章第 2 问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮的时候让地图居中显示某个位置，例如：让地图立即显示第三级居中显示位置（114.2，30.5），需要进行以下几步操作：首先，添加一个 onclick() 函数，用来处理按钮的鼠标点击事件，这个函数调用 IMSMap 控件的 setCenter 函数，传入参数为(114.2,30.5,3)，表示让地图在第三级居中显示(114.2,30.5)；然后给 Button 控件的 click 属性赋值为 onclick()。切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMMap id="map"
        x="34"
        y="81"
```

```

        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
<ns1:TileLayer hdfName="whmap.hdf"
        serverAddress="127.0.0.1:5142">
</ns1:TileLayer>
</ns1:IMSMap>
<mx:Button x="160" y="34" label="居中" click="onclick()" />
<mx:Script>
    <![CDATA[
        public function onclick():void
        {
            this.map.setCenter(114.2,30.5,3);
        }
    ]]>
</mx:Script>
</mx:Application>

```

## 第5问 怎样移动地图至某一位置

首先参照第二章第2问实现瓦片地图显示，然后创建一个 **Button** 控件，如果要想在鼠标点击这个按钮的时候让地图移动到某个位置，例如：让地图立即移动到位置(114.2, 30.5)，需要进行以下几步操作：首先，添加一个 **onclick()**函数，用来处理按钮的鼠标点击事件，这个函数调用 **IMSMap** 控件的 **panTo** 函数，传入参数为(114.2,30.5)，表示让地图居中显示(114.2, 30.5)；然后给 **Button** 控件的 **click** 属性赋值为 **onclick()**。切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">

```

```

<ns1:TileLayer hdfName="whmap.hdf"
    serverAddress="127.0.0.1:5142">
</ns1:TileLayer>
</ns1:IMSMMap>
<mx:Button x="160" y="34" label="移动到(114.2,30.5)"
click="onclick()" />
<mx:Script>
<![CDATA[
    public function onclick():void
    {
        this.map.panTo(114.2,30.5);
    }
}]>
</mx:Script>
</mx:Application>

```

## 第6问 怎样控制地图窗器中的图层是否显示

如果想改变地图容器中的地图状态，需要设置图层控件的 `display` 属性，当其值为 `true` 时，图层处于可见状态，每次操作地图时，显示数据都会更新，否则图层处理隐藏状态并且不更新。以控制一个瓦片图层显示状态为例，先参照第二章第2问实现瓦片地图显示，然后创建一个 `Button` 控件，如果要想在鼠标点击这个按钮的时候切换瓦片图层的显示状态，需要进行以下几步操作：首先，添加一个 `onclick()` 函数，用来处理按钮的鼠标点击事件，这个函数通过设置 `TileLayer` 控件的 `display` 属来控制瓦片图层显示状态；然后给 `Button` 控件的 `click` 属性赋值为 `onclick()`。切换到源码显示窗口，显示如下：图形绘制篇

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*">
    <ns1:IMSMMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMMap>
    <mx:Button x="160" y="34" label="切换显示状态" click="onclick()" />

```

```

<mx:Script>
    <![CDATA[
        public function onclick():void
        {
            this.tileLayer.display=!this.tileLayer.display;
        }
    ]]>
</mx:Script>
</mx:Application>

```

## 第四章 图形绘制篇

### 第1问 怎样在地图上绘制点

首先参照第二章第2问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮时设置鼠标在地图上进行绘制点操作，需要进行以下几步操作：首先，添加一个 onclick() 函数，用来处理按钮的鼠标点击事件，这个函数调用 IMSMap 控件的 setCurOper 函数，传入参数 IMSOperType.SelectByHitting，表示在地图上的鼠标操作状态为点击选取操作，再调用 IMSMap 的 addEventListener 函数给地图上的鼠标弹起事件添加监听，传入参数为 (IMSMapMouseEvent.MAP\_MOUSEUP, drawPoint)，其中 drawPoint 是鼠标弹起事件的处理函数，该函数首先从 IMSMap 中获取鼠标选中的点位置坐标信息，然后创建一个 IMSPoint 对象并将坐标信息赋给它；最后将 IMSPoint 对象作为子控件添加到 IMSMap 中即可。

注意：该示例代码中对 IMSMap 的 IMSMapEvent.MAP\_OPERTYPECHANGE 事件添加了监听，当地图上的鼠标操作切换到其它操作时，将移除绘制点的事件监听，以正常进行其它地图操作。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
xmlns:ns2="zdims.draw.*">
    <ns1:IMSMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"

```



```

        serverAddress="127.0.0.1:5142">
    </ns1:TileLayer>
</ns1:IMSTMap>
<mx:Button x="160" y="34" label="画点" click="onclick()" />
<mx:Script>
    <![CDATA[
        import Mapgis7.WebService.BasLib.Dot_2D;
        import zdims.control.IMSPoint;
        import zdims.event.IMSTMapMouseEvent;
        import zdims.event.IMSTMapEvent;
        import zdims.util.IMSOperType;
        public function init():void
        {
            this.map.addEventListener(IMSTMapEvent.MAP_OPERTYPECHANGE,onOperChange);
        }
        public function onOperChange(e:Event):void
        {
            this.map.removeEventListener(IMSTMapMouseEvent.MAP_MOUSEUP,drawPoint);
        }
        public function onclick():void
        {
            this.map.setCurOper(IMSOperType.SelectByHitting);

            this.map.addEventListener(IMSTMapMouseEvent.MAP_MOUSEUP,drawPoint);
        }
        public function drawPoint(e:Event):void
        {
            var
center:Dot_2D=this.map.graphicCanvas.getSelectedDot();
            var pnt:IMSPoint=new IMSPoint();
            pnt.logicX=center.x;
            pnt.logicY=center.y;
            this.map.addChild(pnt);
        }
    ]]>
</mx:Script>
</mx:Application>

```

## 第2问 怎样在地图上绘制线

首先参照第二章第 2 问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮时设置鼠标在地图上进行绘制线操作，需要进行以下几步操作：首先，添加一个 onclick()函数，用来处理按钮的鼠标双击结束事件，这个函数调用 IMSMap 控件的 setCurOper 函数，传入参数 IMSOperType.SelectByLine，表示在地图上的鼠标操作状态为线选取操作，再调用 IMSMap 的 addEventListener 函数给地图上的鼠标双击事件添加监听，传入参数为(IMSMouseEvent. MAP\_DOUBLECLICK,draw)，其中 draw 是鼠标双击事件的处理函数，该函数首先从 IMSMap 中获取鼠标点击操作选取的所有点位置坐标信息，然后创建一个 IMSRoad 对象并将坐标信息赋给它；最后将 IMSRoad 对象作为子控件添加到 IMSMap 中即可。

注意：该示例代码中对 IMSMap 的 IMSMapEvent.MAP\_OPERTYPECHANGE 事件添加了监听，当地图上的鼠标操作切换到其它操作时，将移除绘制线的事件监听，以正常进行其它地图操作。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
xmlns:ns2="zdims.draw.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMAP>
    <mx:Button x="160" y="34" label="画线" click="onclick()" />
    <mx:Script>
        <![CDATA[
            import zdims.control.IMSRoad;
            import Mapgis7.WebService.BasLib.Dot_2D;
            import zdims.event.IMSMouseEvent;
            import zdims.event.IMSMapEvent;
            import zdims.util.IMSOperType;
            public function init():void
            {
```

```

        this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE, onOperChange);
    }
    public function onOperChange(e:Event):void
    {

        this.map.removeEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK, draw);
    }
    public function onclick():void
    {
        this.map.setCurOper(IMSOperType.SelectByLine);

        this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK, draw);
    }
    public function draw(e:Event):void
    {
        var
dots:Array=this.map.graphicCanvas.getSelectedLogicDotsArr();
        var line:IMSRoad=new IMSRoad();
        line.setDots(dots);
        this.map.addChild(line);
    }
    ]]>
</mx:Script>
</mx:Application>

```

### 第3问 怎样在地图上绘制圆

在地图上绘制圆跟第四章第1问讲述的绘制点类似，参考示例代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
xmlns:ns2="zdims.draw.*">
    <ns1:IMSMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"

```

```

        yMaxMap="30.8291188260302">
<ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
        serverAddress="127.0.0.1:5142">
</ns1:TileLayer>
</ns1:IMSMap>
<mx:Button x="160" y="34" label="画圆" click="onclick()" />
<mx:Script>
    <![CDATA[
        import Mapgis7.WebService.BasLib.Circle;
        import Mapgis7.WebService.BasLib.Dot_2D;
        import zdims.control.IMSPoint;
        import zdims.event.IMSMapMouseEvent;
        import zdims.event.IMSMapEvent;
        import zdims.util.IMSOperType;
        public function init():void
        {

            this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE,onOperCh
ange);
        }
        public function onOperChange(e:Event):void
        {

            this.map.removeEventListener(IMSMapMouseEvent.MAP_MOUSEUP,drawPoi
nt);
        }
        public function onclick():void
        {
            this.map.setCurOper(IMSOperType.SelectByCircle);

            this.map.addEventListener(IMSMapMouseEvent.MAP_MOUSEUP,drawPoint)
;
        }
        public function drawPoint(e:Event):void
        {
            var
circle:Circle=this.map.graphicCanvas.getSelectedCircle();
            var pnt:IMSPoint=new IMSPoint();
            pnt.logicX=circle.Center.x;
            pnt.logicY=circle.Center.y;
            pnt.radius=circle.Radius;
            pnt.enableUpadateRadius=true;
            this.map.addChild(pnt);
        }
    ]]>

```

```

]]>
</mx:Script>
</mx:Application>

```

## 第4问 怎样在地图上绘制矩形

首先参照第二章第 2 问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮时设置鼠标在地图上进行绘制矩形操作，需要进行以下几步操作：首先，添加一个 onclick() 函数，用来处理按钮的鼠标弹起事件，这个函数调用 IMSMap 控件的 setCurOper 函数，传入参数 IMSOperType.SelectByRect，表示在地图上的鼠标操作状态为矩形区域选取操作，再调用 IMSMap 的 addEventListener 函数给地图上的鼠标弹起事件添加监听，传入参数为 (IMSMapMouseEvent, MAP\_MOUSEUP, draw)，其中 draw 是鼠标弹起事件的处理函数，该函数首先从 IMSMap 中获取鼠标操作选取的矩形范围对象，然后创建一个 IMSRoad 对象并将矩形范围赋给它；最后将 IMSRoad 对象作为子控件添加到 IMSMap 中即可。

注意：该示例代码中对 IMSMap 的 IMSMapEvent.MAP\_OPERTYPECHANGE 事件添加了监听，当地图上的鼠标操作切换到其它操作时，将移除绘制矩形的事件监听，以正常进行其它地图操作。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
xmlns:ns2="zdims.draw.*">
  <ns1:IMSMap id="map"
    x="34"
    y="81"
    width="703"
    height="535"
    levelNum="7"
    xMinMap="114.125602229914"
    xMaxMap="114.500788705197"
    yMinMap="30.4539323507469"
    yMaxMap="30.8291188260302">
    <ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
      serverAddress="127.0.0.1:5142">
    </ns1:TileLayer>
  </ns1:IMSMap>
  <mx:Button x="160" y="34" label="画矩形" click="onclick()" />
  <mx:Script>
    <![CDATA[
      import zdims.control.IMSRoad;
      import Mapgis7.WebService.BasLib.Rect;
      import zdims.event.IMSMapMouseEvent;

```

```

import zdims.event.IMSMapEvent;
import zdims.util.IMSOperType;
public function init():void
{

    this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE,onOperChange);

    public function onOperChange(e:Event):void
    {

        this.map.removeEventListener(IMSMapMouseEvent.MAP_MOUSEUP,draw);

        public function onclick():void
        {

            this.map.setCurOper(IMSOperType.SelectByRect);

            this.map.addEventListener(IMSMapMouseEvent.MAP_MOUSEUP,draw);

            public function draw(e:Event):void
            {

                var
selRect:Rect=this.map.graphicCanvas.getSelectedRect();
                var rect:IMSRoad=new IMSRoad();
                rect.enableFill=true;
                rect.setDotsFromRect(selRect);
                this.map.addChild(rect);

            }
        }
    }
}
</mx:Script>
</mx:Application>

```

## 第5问 怎样在地图上绘制区

在地图上绘制区跟本章第 2 问类似，不同的是绘制区需要进行填充即将 `IMSRoad` 的 `enableFill` 属性设置为 `true`，另外在调用 `setCurOper` 函数设置地图操作类型时传入的参数是 `(IMSOperType.SelectByPolygon)` 表示鼠标操作类型是多边形选取操作。参考示例代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
xmlns:ns2="zdims.draw.*">
    <ns1:IMSMap id="map"
        x="34"

```

```

        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
<ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
        serverAddress="127.0.0.1:5142">
</ns1:TileLayer>
</ns1:IMSTMap>
<mx:Button x="160" y="34" label="画多边形" click="onclick()"/>
<mx:Script>
    <![CDATA[
        import zdims.control.IMSRoad;
        import zdims.event.IMSTMapMouseEvent;
        import zdims.event.IMSTMapEvent;
        import zdims.util.IMSOperType;
        public function init():void
        {

            this.map.addEventListener(IMSTMapEvent.MAP_OPERTYPECHANGE,onOperChange);

        }
        public function onOperChange(e:Event):void
        {

            this.map.removeEventListener(IMSTMapMouseEvent.MAP_DOUBLECLICK,draw);

        }
        public function onclick():void
        {
            this.map.setCurOper(IMSOperType.SelectByPolygon);

            this.map.addEventListener(IMSTMapMouseEvent.MAP_DOUBLECLICK,draw);
        }
        public function draw(e:Event):void
        {
            var
dots:Array=this.map.graphicCanvas.getSelectedLogicDotsArr();
            var reg:IMSRoad=new IMSRoad();
            reg.setDots(dots);
            reg.enableFill=true;

```

```

        this.map.addChild(reg);
    }
}]>
</mx:Script>
</mx:Application>

```

## 第6问 怎样在地图上添加一个点

第四章第1问介绍了怎么在地图上通过鼠标绘制一个点在地图上叠加显示,如果在实际应用中不需要通过鼠标交互来绘制点,而是已经知道点的位置了,想要立即在地图上添加一个点,这种情况下可以通过代码快速的在地图上添加一个点,只需先创建一个 `IMSPoint` 对象并给其 `logicX` 和 `logicY` 坐标赋值,然后将其作为子节点添加到 `IMSMap` 中即可。下面的示例代码实现了点击一个按钮的时候立即在坐标 (114.2, 30.5) 添加一个点图形,让其叠加在地图上。示例代码如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*" xmlns:ns2="zdims.draw.*"
fontSize="12">
    <ns1:IMSMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMap>
    <mx:Button x="160" y="34" label="添加点" click="onclick()"/>
    <mx:Script>
    <![CDATA[
        import Mapgis7.WebService.BasLib.Dot_2D;
        import zdims.control.IMSPoint;
        public function onclick():void
        {
            var pnt:IMSPoint=new IMSPoint();
            pnt.logicX=114.2;
            pnt.logicY=30.5;
            this.map.addChild(pnt);
        }
    ]]>
    </mx:Script>

```



```

    }
  ]]>
</mx:Script>
</mx:Application>

```

## 第7问 怎样在地图上添加一条线

第四章第2问介绍了怎么在地图上通过鼠标绘制一条线在地图上叠加显示,如果在实际应用中不需要通过鼠标交互来绘制点,而是已经知道线上的各个点的位置了,想要立即在地图上添加一条线,这种情况下可以通过代码快速的在地图上添加一条线,只需先创建一个 `IMSRoad` 对象并调用其 `setDots` 函数将 `Point` 点数组设置给 `IMSRoad`, 然后将其作为子节点添加到 `IMSMap` 中即可。下面的示例代码实现了点击一个按钮的时候立即把三个坐标 (114.2, 30.5), (114.3, 30.5), (114.25, 30.6) 连接成一条线, 让其叠加在地图上显示。

**提示:** 在地图上添加一个多边形跟添加一条线类似, 只需要将 `IMSRoad` 的 `enableFill` 置为 `true` 即可。

示例代码如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*" xmlns:ns2="zdims.draw.*"
fontSize="12">
  <ns1:IMSMap id="map"
    x="34"
    y="81"
    width="703"
    height="535"
    levelNum="7"
    xMinMap="114.125602229914"
    xMaxMap="114.500788705197"
    yMinMap="30.4539323507469"
    yMaxMap="30.8291188260302">
    <ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
      serverAddress="127.0.0.1:5142">
    </ns1:TileLayer>
  </ns1:IMSMap>
  <mx:Button x="160" y="34" label="添加线" click="onclick()" />
  <mx:Script>
    <![CDATA[
      import Mapgis7.Webservice.BasLib.Dot_2D;
      import zdims.control.IMSPoint;
      public function onclick():void
      {
        var dots:Array=new Array();
        dots[0]=new Point(114.2,30.5);

```

```

        dots[1]=new Point(114.3,30.5);
        dots[2]=new Point(114.25,30.6);
        var line:IMSRoad=new IMSRoad();
        line.setDots(dots);
        this.map.addChild(line);
    }
]]>
</mx:Script>
</mx:Application>

```

## 第8问 怎样在地图上添加一个标注点

首先参照第二章第2问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮时在地图上添加一个标注，需要进行以下几步操作：首先，添加一个 onclick() 函数，用来处理按钮的鼠标点击事件，在该函数中先创建一个 Marker 对象并将坐标信息赋给它；然后将 Marker 对象作为子控件添加到 IMSMap 中即可。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*" xmlns:ns2="zdims.draw.*"
fontSize="12">
    <ns1:IMSMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMap>
    <mx:Button x="160" y="34" label="添加标注点" click="onclick()" />
    <mx:Script>
        <![CDATA[
            import zdims.control.Marker;
            public function onclick():void
            {
                var marker:Marker=new Marker();
                marker.logicX=114.2;
            }
        ]]>
    </mx:Script>

```

```

        marker.logicY=30.5;
        marker.setIconSrc("image/mark/p2.gif");
        this.map.addChild(marker);
    }
}]>
</mx:Script>
</mx:Application>

```

## 第9问 怎样实现地图窗口坐标和逻辑坐标之间的转换

首先参照第二章第2问实现瓦片地图显示，然后创建一个 Button 控件，下面的示例代码实现了鼠标点击这个按钮的时候调用 IMSMap 中的 screenToLogic 函数把窗口坐标（200, 200）转换为逻辑坐标，然后调用 logicToScreen 把得到的逻辑坐标再反转成窗口坐标。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*" xmlns:ns2="zdims.draw.*"
fontSize="12">
    <ns1:IMSMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMap>
    <mx:Button x="160" y="34" label="坐标转换" click="onclick()"/>
    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;
            public function onclick():void
            {
                var pnt:Point=this.map.screenToLogic(200,200);
                Alert.show("窗口坐标【200, 200】对应的逻辑坐标是
                ["+pnt.x+", "+pnt.y+"]");
                var scrPnt:Point=this.map.logicToScreen(pnt.x,pnt.y);
                Alert.show("逻辑坐标【"+pnt.x+", "+pnt.y+"】对应的窗口坐标是

```

```

【"+scrPnt.x+","+scrPnt.y+"】");
    }
    ]]>
</mx:Script>
</mx:Application>

```

## 第10问 怎样将统计图显示在地图上并跟随地图放大缩小

首先参照第二章第2问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮时在地图上添加一个统计图（以添加柱状统计图为例），需要进行以下几步操作：首先，添加一个 onclick() 函数，用来处理按钮的鼠标点击事件，在该函数中先创建一个 IMSColumnChart 对象并将统计数据 and 坐标信息赋给它，将其 enableUpdatePosition 属性置为 true 以让其跟随地图移动；然后将 IMSColumnChart 对象作为子控件添加到 IMSMap 中即可。

**注意：**本示例监听的是 IMSMap 的 MAP\_CHANGE 事件，在事件处理函数中获取的是瓦片的显示级别，然后设置柱状图的显示比例，如果在实际应用中 IMSMap 中没有瓦片图层，则需要采用另外的方法来控件柱状图的显示比例。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*" xmlns:ns2="zdims.draw.*"
fontSize="12" xmlns:ns3="zdims.control.*">
    <ns1:IMSMap id="map"
        x="34"
        y="81"
        width="703"
        height="535"
        levelNum="7"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:TileLayer id="tileLayer" hdfName="whmap.hdf"
            serverAddress="127.0.0.1:5142">
        </ns1:TileLayer>
    </ns1:IMSMap>
    <mx:Button x="160" y="34" label="添加统计图" click="onclick()"/>
    <mx:Script>
        <![CDATA[
            import zdims.event.IMSMapZoomEvent;
            import zdims.event.IMSMapEvent;
            import zdims.control.IMSColumnChart;
            public function onclick():void

```

```

    {
        var columnChart:IMSColumnChart=new IMSColumnChart();

        columnChart.dataProvider=[{x:"x1",y:20},{x:"x2",y:30},{x:"x3",y:25}];

        columnChart.enableUpdatePosition=true;
        columnChart.logicX=114.2;
        columnChart.logicY=30.6;
        this.map.addChild(columnChart);

        this.map.addEventListener(IMSMapEvent.MAP_CHANGE,function(e:Event):void{onChange(columnChart)});
    }
    public function onChange(chart:IMSColumnChart):void
    {

        chart.scaleX=chart.scaleY=this.map.getLevel()==0?1:this.map.getLevel();
    }
}]>
</mx:Script>
</mx:Application>

```

## 第11问 怎样添加一个热点区域

首先参照第二章第2问实现瓦片地图显示，然后创建一个 Button 控件，如果要想在鼠标点击这个按钮时在地图上添加一个热点区域，需要进行以下几步操作：首先，添加一个 onclick()函数，用来处理按钮的鼠标点击事件，在该函数中先创建一个 HotSpot 对象并将热点区域坐标信息赋给它；然后将 HotSpot 对象作为子控件添加到 IMSMap 中即可。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
xmlns:ns2="zdims.control.*" xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMap id="map" levelNum="7"
        x="41"
        y="40"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">

```

```

        <ns1:TileLayer hdfName="whmap.hdf">

        </ns1:TileLayer>
    </ns1:IMSMap>
    <mx:Button x="76" y="10" label="添加热点" click="onClick()" />
    <mx:Script>
        <![CDATA[
            import zdims.control.HotSpot;
            public function onClick():void
            {
                var hot:HotSpot=new HotSpot();
                hot.hotLabelText="鼠标压到我身上，好不舒服啊！";
                hot.hotAreaDots=[new Point(114.2,30.5),new
Point(114.3,30.5),new Point(114.3,30.6),new Point(114.2,30.5)];
                this.map.addChild(hot);
            }

        ]]>
    </mx:Script>

</mx:Application>

```

## 第五章 开发函数篇

### 第1问 怎样获取地图文档图层列表

参照第二章第 3 问实现地图文档显示，然后添加一个按钮，在点击该按钮的时候触发 onClick 函数。在 onClick 函数中通过调用地图文档对象 VectorMapDoc 的 getLayerInfo 函数获取地图文档中的各个图层名称。切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*" fontSize="12">
    <ns1:IMSMap id="map"
        x="34"
        y="54"
        width="703"
        height="562"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">

```

```

        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSTMap>
    <mx:Button x="209" y="24" label="获取图层" click="onClick()" />
    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;
            public function onClick():void
            {
                var
layerCount:int=this.mapdoc.loadMapResult.LayerCount;
                var layers:Array=new Array();
                for(var i:int=0;i<layerCount;i++)
                {

layers[i]=this.mapdoc.getLayerInfo(i).LayerDataName;
                }
                Alert.show("wh.map中有这些图层: "+layers.join(", "));
            }
        ]]>
    </mx:Script>
</mx:Application>

```

## 第2问 怎样更新地图文档中的图层的状态

参照第二章第 3 问实现地图文档显示, 然后添加一个按钮, 在点击该按钮的时候触发 onClick 函数。在 onClick 函数中通过调用地图文档对象 VectorMapDoc 的 updateAllLayerInfo 更新地图文档中的图层显示状态。切换到源码显示窗口, 显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute" xmlns:ns1="zdims.map.*" fontSize="12">
    <ns1:IMSTMap id="map"
        x="34"
        y="54"
        width="703"
        height="562"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"

```

```

        mapDocName="wh.map"
        serverAddress="127.0.0.1:5141">
    </ns1:VectorMapDoc>
</ns1:IMSMap>
<mx:Button x="209" y="24" label="设置第0个图层不可见"
click="onClick()" />
<mx:Script>
    <![CDATA[
        import Mapgis7.WebService.BasLib.EnumLayerStatus;
        import mx.controls.Alert;
        public function onClick():void
        {

            this.mapdoc.getMapLayerInfo(0).LayerStatus=EnumLayerStatus.Invisi
able;

            this.mapdoc.updateAllLayerInfo();
            Alert.show("第0个图层已设置为不可见，刷新一下地图试试效果");
        }
    ]]>
</mx:Script>
</mx:Application>

```

### 第3问 怎样获取鼠标当前位置信息

参照第二章第3问实现地图文档显示，然后 Application 创建完成的时候给地图添加鼠标移动事件监听，在事件处理函数中获取 IMSMap 的 mouseMoveLogicPnt 属性从而得到鼠标所在位置的坐标，然后设置在 Label 控件中显示。切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12">
    <ns1:IMSMap id="map"
        x="34"
        y="54"
        width="703"
        height="562"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">

```



```

        </ns1:VectorMapDoc>
    </ns1:IMSTMap>
    <mx:Label id="position" x="135" y="26" text="鼠标在地图上的位置"/>
    <mx:Script>
        <![CDATA[
            import zdims.event.IMSTMapMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSTMapMouseEvent.MAP_MOUSEMOVE, onMouseMove);

            }
            public function onMouseMove(e:Event):void
            {
                this.position.text="鼠标在地图上的位置:
"+this.map.mouseMoveLogicPnt.x+", "+this.map.mouseMoveLogicPnt.y;
            }
        ]]>
    </mx:Script>

</mx:Application>

```

## 第4问 怎样添加和删除地图鼠标事件

参照本章第3问实现鼠标位置获取功能, Application 创建完成的时候给地图添加鼠标移动事件监听, 添加一个 CheckBox 控件, 给其 change 事件添加监听, 在事件处理函数 onChange 函数中判断该控件是否处于选中状态, 如果被选中, 则添加鼠标事件监听, 如果没选中, 则移除鼠标事件监听。切换到源码显示窗口, 显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12">
    <ns1:IMSTMap id="map"
        x="34"
        y="54"
        width="703"
        height="562"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"

```

```

serverAddress="127.0.0.1:5141">
</ns1:VectorMapDoc>
</ns1:IMSMap>
<mx:Label id="position" x="223" y="12" text="鼠标在地图上的位置"/>
<mx:Script>
<![CDATA[
import zdims.event.IMSMapMouseEvent;
public function init():void
{

this.map.addEventListener(IMSEMapMouseEvent.MAP_MOUSEMOVE,onMouseM
ove);

}
public function onMouseMove(e:Event):void
{

this.position.text="鼠标在地图上的位置:
"+this.map.mouseMoveLogicPnt.x+", "+this.map.mouseMoveLogicPnt.y;
}
public function onChange(e:Event):void
{

if(e.target.selected)

this.map.addEventListener(IMSEMapMouseEvent.MAP_MOUSEMOVE,onMouseM
ove);

else

this.map.removeEventListener(IMSEMapMouseEvent.MAP_MOUSEMOVE,onMou
seMove);

}
}]>
</mx:Script>
<mx:CheckBox x="57" y="10" selected="true" change="onChange(event)"
label="是否监听鼠标事件"/>

</mx:Application>

```

## 第5问 怎样用鼠标获取一个点

参照第二章第3问实现地图文档显示，然后 Application 创建完成的时候给地图添加鼠标单击事件监听，在事件处理函数中获取 IMSMap 的 mouseDownLogicPnt 属性从而得到鼠标所在位置的坐标，然后设置在 Label 控件中显示。切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"

```

```

creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12">
    <ns1:IMSTMap id="map"
        x="34"
        y="54"
        width="703"
        height="562"
        xminMap="114.125602229914"
        xmaxMap="114.500788705197"
        yminMap="30.4539323507469"
        ymaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSTMap>
    <mx:Label id="position" x="223" y="12" text="鼠标单击选中的位置: "/>
    <mx:Script>
        <![CDATA[
            import zdims.event.IMSTMapMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSTMapMouseEvent.MAP_CLICK,onMouseClick
            );

            }
            public function onMouseClick(e:Event):void
            {
                this.position.text="鼠标单击选中的位置:
            "+this.map.mouseDownLogicPnt.x+", "+this.map.mouseDownLogicPnt.y;
            }
        ]]>
    </mx:Script>

</mx:Application>

```

## 第6问 怎样用鼠标获取一条线

参照第二章第 3 问实现地图文档显示, 然后 Application 创建完成的时候给地图添加鼠标双击事件监听, 在事件处理函数中获取 IMSTMap 的 graphicCanvas 属性的 getSelectedLogicDotsArr 函数获取鼠标绘制的线上的点坐标集合, 然后设置在 TextArea 控件中显示。切换到源码显示窗口, 显示如下:

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12">
    <ns1:IMSMAP id="map"
        x="34"
        y="139"
        width="703"
        height="477"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMAP>
    <mx:TextArea id="position" x="49" y="12" text="点坐标序列: "
width="650" height="119"/>
    <mx:Script>
        <![CDATA[
            import zdims.util.IMSOperType;
            import zdims.event.IMSMapMouseEvent;
            public function init():void
            {
                this.map.setCurOper(IMSOperType.SelectByLine);

                this.map.addEventListener(IMSMAPMouseEvent.MAP_DOUBLECLICK, callBa
ck);
            }
            public function callBack(e:Event):void
            {
                var
dots:Array=this.map.graphicCanvas.getSelectedLogicDotsArr();
                this.position.text="点坐标序列: "+dots.join(",");
            }
        ]]>
    </mx:Script>

</mx:Application>

```

## 第7问 怎样用鼠标获取一个矩形

参照第二章第 3 问实现地图文档显示，然后 Application 创建完成的时候给地图添加鼠标弹起事件监听，在事件处理函数中获取 IMSMap 的 graphicCanvas 属性的 getSelectedRect 函数获取鼠标绘制的矩形，然后设置在 TextArea 控件中显示矩形左下角和右上角坐标。切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12">
    <ns1:IMSMap id="map"
        x="34"
        y="139"
        width="703"
        height="477"
        xminMap="114.125602229914"
        xmaxMap="114.500788705197"
        yminMap="30.4539323507469"
        ymaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMap>
    <mx:TextArea id="position" x="49" y="12" text="矩形范围: " width="650"
height="119"/>
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.Rect;
            import zdims.util.IMSOperType;
            import zdims.event.IMSMapMouseEvent;
            public function init():void
            {
                this.map.setCurOper(IMSOperType.SelectByRect);

                this.map.addEventListener(IMSMapMouseEvent.MAP_MOUSEUP, callBack);
            }
            public function callBack(e:Event):void
            {
                var rect:Rect=this.map.graphicCanvas.getSelectedRect();
                this.position.text="矩形范围:
xmin="+rect.xmin+",ymin="+rect.ymin+",xmax="+rect.xmax+",ymax="+rect.
ymax;
            }
        ]]>
    </mx:Script>
</mx:Application>
```

```

]]>
</mx:Script>

</mx:Application>

```

## 第8问 怎样用鼠标获取一个圆

参照第二章第 3 问实现地图文档显示，然后 Application 创建完成的时候给地图添加鼠标弹起事件监听，在事件处理函数中获取 IMSMap 的 graphicCanvas 属性的 getSelectedCircle 函数获取鼠标绘制的圆，然后设置在 TextArea 控件中显示圆心和半径。切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12">
    <ns1:IMSMap id="map"
        x="34"
        y="139"
        width="703"
        height="477"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMap>
    <mx:TextArea id="position" x="49" y="12" text="圆: " width="650"
height="119"/>
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.Circle;
            import Mapgis7.WebService.BasLib.Rect;
            import zdims.util.IMSOperType;
            import zdims.event.IMSMapMouseEvent;
            public function init():void
            {
                this.map.setCurOper(IMSOperType.SelectByCircle);

                this.map.addEventListener(IMSMapMouseEvent.MAP_MOUSEUP, callBack);
            }
        ]]>
    </mx:Script>

```

```

        public function callBack(e:Event):void
        {
            var
obj:Circle=this.map.graphicCanvas.getSelectedCircle();
            this.position.text="圆心
("+obj.Center.x+","+obj.Center.y+") 半径="+obj.Radius;
        }
    ]]>
</mx:Script>

</mx:Application>

```

## 第9问 怎样用鼠标获取一个多边形

参照第二章第 3 问实现地图文档显示, 然后 Application 创建完成的时候给地图添加鼠标双击事件监听, 在事件处理函数中获取 IMSMap 的 graphicCanvas 属性的 getSelectedPolygon 函数获取鼠标绘制的多边形, 然后设置在 TextArea 控件中显示多边形上的各个结点的坐标。切换到源码显示窗口, 显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12">
    <ns1:IMSMap id="map"
        x="34"
        y="139"
        width="703"
        height="477"
        xminMap="114.125602229914"
        xmaxMap="114.500788705197"
        yminMap="30.4539323507469"
        ymaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMap>
    <mx:TextArea id="position" x="49" y="12" text="多边形上的点序列: "
width="650" height="119"/>
    <mx:Script>
        <![CDATA[
            import Mapgis7.Webservice.BasLib.Polygon;
            import mx.controls.Alert;
            import Mapgis7.Webservice.BasLib.Circle;

```

```

import Mapgis7.WebService.BasLib.Rect;
import zdims.util.IMSOperType;
import zdims.event.IMSMapMouseEvent;
public function init():void
{
    this.map.setCurOper(IMSOperType.SelectByPolygon);

    this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK, callBa
ck);
}
public function callBack(e:Event):void
{
    var
obj:Polygon=this.map.graphicCanvas.getSelectedPolygon();
    if(obj==null)
        return;
    var coor:String="多边形上的点序列: ";
    for(var i:int=0;i<obj.Dots.length;i++)
    {
        coor+="(x="+obj.Dots[i].x+",y="+obj.Dots[i].y+") ";
    }
    this.position.text=coor;
}
]]>
</mx:Script>
</mx:Application>

```

## 第10问 怎样实现点击查询

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第0个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByHitting，同时监听地图的单击事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedDot 方法得到鼠标选取的点，然后创建查询参数对象实例并给其相关参数赋值，最后调用 VectorMapDoc 的 select 方法实现点击查询。

**注意：**为了方便展示查询结果，本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以显示查询结果。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMap id="map"

```



```

        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
<ns1:VectorMapDoc id="mapdoc"
        mapDocName="wh.map"
        serverAddress="127.0.0.1:5141">
</ns1:VectorMapDoc>
<ns2:MapDocDataViewer id="dataViewer" imsmap="{map}">
</ns2:MapDocDataViewer>
</ns1:IMSMMap>
<mx:Button x="76" y="10" label="设置第0个图层为可查询状态"
click="onClick()" />
<mx:Script>
    <![CDATA[
        import zdims.event.IMSMMapEvent;
        import zdims.util.IMSOperType;
        import Mapgis7.WebService.BasLib.EnumLayerStatus;
        import Mapgis7.WebService.BasLib.CMapSelectParam;
        import Mapgis7.WebService.BasLib.EWebGeomType;
        import Mapgis7.WebService.BasLib.ESelectionType;
        import Mapgis7.WebService.BasLib.CWebSelectParam;
        import zdims.event.IMSMMapMouseEvent;
        public function init():void
        {

            this.map.addEventListener(IMSMMapEvent.MAP_OPERTYPECHANGE,onOperCh
ange);
        }
        public function onOperChange(e:Event):void
        {

            this.map.removeEventListener(IMSMMapMouseEvent.MAP_CLICK,onMouseCl
ick);
        }
        public function onMouseClick(e:Event):void
        {
            var obj:CWebSelectParam=new CWebSelectParam();
            obj.SelectionType=ESelectionType.SpatialRange;
            obj.GeomType=EWebGeomType.Point;

```

```

        obj.Geometry=this.map.graphicCanvas.getSelectedDot();
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param,callback);
    }
    public function callBack(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        dataViewer.showResult(this.mapdoc.onSelect(e));
    }
    public function onClick():void
    {
        this.mapdoc.getMapLayerInfo(0).LayerStatus=EnumLayerStatus.Selectable;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByHitting);

        this.map.addEventListener(IMSMapMouseEvent.MAP_CLICK,onMouseClick);
    }
]]>
</mx:Script>

</mx:Application>

```

## 第11问 怎样实现线查询

参照第二章第 3 问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 0 个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByLine，同时监听地图的双击事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedLine 方法得到鼠标选取的线对象，然后创建查询参数对象实例并给其相关参数赋值，最后调用 VectorMapDoc 的 select 方法实现线查询。

**注意：**为了方便展示查询结果，本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以显示查询结果。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMap id="map"
        x="34"

```

```

        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
<ns1:VectorMapDoc id="mapdoc"
        mapDocName="wh.map"
        serverAddress="127.0.0.1:5141">
</ns1:VectorMapDoc>
<ns2:MapDocDataViewer id="dataViewer" imsmap="{map}">
</ns2:MapDocDataViewer>
</ns1:IMSMAP>
<mx:Button x="76" y="10" label="设置第0个图层为可查询状态"
click="onClick()" />
<mx:Script>
    <![CDATA[
        import Mapgis7.WebService.BasLib.IWebGeometry;
        import zdims.event.IMSMapEvent;
        import zdims.util.IMSOperType;
        import Mapgis7.WebService.BasLib.EnumLayerStatus;
        import Mapgis7.WebService.BasLib.CMapSelectParam;
        import mx.controls.Alert;
        import Mapgis7.WebService.BasLib.EWebGeomType;
        import Mapgis7.WebService.BasLib.ESelectionType;
        import Mapgis7.WebService.BasLib.CWebSelectParam;
        import zdims.event.IMSMapMouseEvent;
        public function init():void
        {

            this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE,onOperCh
ange);
        }
        public function onOperChange(e:Event):void
        {

            this.map.removeEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,mou
seCallBack);
        }
        public function mouseCallBack(e:Event):void
        {
            var obj:CWebSelectParam=new CWebSelectParam();
            obj.SelectionType=ESelectionType.SpatialRange;

```

```

        obj.Geometry=this.map.graphicCanvas.getSelectedLine();
        if(obj.Geometry==null)
            return;
        obj.GeomType=EWebGeomType.Line;
        obj.NearDistance=0.0001;
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param,callback);
    }
    public function callBack(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        dataViewer.showResult(this.mapdoc.onSelect(e));
    }
    public function onClick():void
    {
        this.mapdoc.getMapLayerInfo(0).LayerStatus=EnumLayerStatus.
        Selectable;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByLine);

        this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,mouseC
        allBack);
    }
    ]]>
</mx:Script>

</mx:Application>

```

## 第12问 怎样实现矩形查询

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 0 个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByRect，同时监听地图的鼠标弹起事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedRect 方法得到鼠标选取的矩形对象，然后创建查询参数对象实例并给其相关参数赋值，最后调用 VectorMapDoc 的 select 方法实现矩形查询。

**注意：**为了方便展示查询结果，本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以显示查询结果。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"

```

```

creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" imsmap="{map}">
        </ns2:MapDocDataViewer>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="设置第0个图层为可查询状态"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.IWebGeometry;
            import zdims.event.IMSMAPEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import mx.controls.Alert;
            import Mapgis7.WebService.BasLib.EWebGeomType;
            import Mapgis7.WebService.BasLib.ESelectionType;
            import Mapgis7.WebService.BasLib.CWebSelectParam;
            import zdims.event.IMSMAPMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSMAPEvent.MAP_OPERTYPECHANGE,onOperCh
ange);

            }
            public function onOperChange(e:Event):void
            {

                this.map.removeEventListener(IMSMAPMouseEvent.MAP_MOUSEUP,mouseCa
llBack);

            }

```

```

public function mouseCallBack(e:Event):void
{
    var obj:CWebSelectParam=new CWebSelectParam();
    obj.SelectionType=ESelectionType.SpatialRange;
    obj.Geometry=this.map.graphicCanvas.getSelectedRect();
    if(obj.Geometry==null)
    {
        return;
    }
    obj.GeomType=EWebGeomType.Rect;
    var param:CMapSelectParam=new CMapSelectParam();
    param.SelectParam=obj;
    this.mapdoc.select(param,callback);
}

public function callBack(e:Event):void
{
    this.map.activeMapDoc=this.mapdoc;
    dataViewer.showResult(this.mapdoc.onSelect(e));
}

public function onClick():void
{
    this.mapdoc.getMapLayerInfo(0).LayerStatus=EnumLayerStatus.Selectable;

    this.mapdoc.updateAllLayerInfo();
    this.map.setCurOper(IMSOperType.SelectByRect);

    this.map.addEventListener(IMSMapMouseEvent.MAP_MOUSEUP,mouseCallBack);
}
}]>
</mx:Script>

</mx:Application>

```

## 第13问 怎样实现圆查询

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 0 个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByCircle，同时监听地图的鼠标弹起事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedCircle 方法得到鼠标选取的圆对象，然后创建查询参数对象实例并给其相关参数赋值，最后调用 VectorMapDoc 的 select 方法实现圆查询。

**注意：**为了方便展示查询结果，本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以显示查询结果。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" imsmap="{map}">
        </ns2:MapDocDataViewer>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="设置第0个图层为可查询状态"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.IWebGeometry;
            import zdims.event.IMSMAPEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import mx.controls.Alert;
            import Mapgis7.WebService.BasLib.EWebGeomType;
            import Mapgis7.WebService.BasLib.ESelectionType;
            import Mapgis7.WebService.BasLib.CWebSelectParam;
            import zdims.event.IMSMAPMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSMAPEvent.MAP_OPERTYPECHANGE,onOperCh
ange);

            }
            public function onOperChange(e:Event):void
            {
```

```

        this.map.removeEventListener(IMSMapMouseEvent.MAP_MOUSEUP, mouseCa
llBack);
    }
    public function mouseCallBack(e:Event):void
    {
        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.SpatialRange;

        obj.Geometry=this.map.graphicCanvas.getSelectedCircle();
        if(obj.Geometry==null)
            return;
        obj.GeomType=EWebGeomType.Circle;
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param, callBack);
    }
    public function callBack(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        dataViewer.showResult(this.mapdoc.onSelect(e));
    }
    public function onClick():void
    {
        this.mapdoc.getMapLayerInfo(0).LayerStatus=EnumLayerStatus.Select
able;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByCircle);

        this.map.addEventListener(IMSMapMouseEvent.MAP_MOUSEUP, mouseCallB
ack);
    }
]]>
</mx:Script>

</mx:Application>

```

## 第14问 怎样实现多边形查询

参照第二章第 3 问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 0 个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByPolygon，同时监听地图的双击事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedPolygon 方法得到鼠标选取的多线形对象，然后创建查询参数对



象实例并给其相关参数赋值，最后调用 `VectorMapDoc` 的 `select` 方法实现多边形查询。

**注意：**为了方便展示查询结果，本示例使用了地图文档查询结果显示控件 `MapDocDataViewer`。在查询的回调函数中调用了 `MapDocDataViewer` 的 `showResult` 函数以显示查询结果。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" imsmap="{map}">
        </ns2:MapDocDataViewer>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="设置第0个图层为可查询状态"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.Webservice.BasLib.IWebGeometry;
            import zdims.event.IMSMAPEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.Webservice.BasLib.EnumLayerStatus;
            import Mapgis7.Webservice.BasLib.CMapSelectParam;
            import mx.controls.Alert;
            import Mapgis7.Webservice.BasLib.EWebGeomType;
            import Mapgis7.Webservice.BasLib.ESelectionType;
            import Mapgis7.Webservice.BasLib.CWebSelectParam;
            import zdims.event.IMSMAPMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSMAPEvent.MAP_OPERTYPECHANGE, onOperCh
ange);
            }
        ]]>
    </mx:Script>
</mx:Application>
```

```
    }
    public function onOperChange(e:Event):void
    {

        this.map.removeEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,mouseCallback);
    }
    public function mouseCallback(e:Event):void
    {
        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.SpatialRange;

        obj.Geometry=this.map.graphicCanvas.getSelectedPolygon();
        if(obj.Geometry==null)
            return;
        obj.GeomType=EWebGeomType.Polygon;
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param,callback);
    }
    public function callback(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        dataViewer.showResult(this.mapdoc.onSelect(e));
    }
    public function onClick():void
    {

        this.mapdoc.getMapLayerInfo(0).LayerStatus=EnumLayerStatus.Selectable;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByPolygon);

        this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,mouseCallback);
    }
]]>
</mx:Script>

</mx:Application>
```

## 第15问 怎样实现属性条件查询

参照第二章第3问实现地图文档显示, 添加一个 Button 控件, 在点击该 Button 时, 设置地图文档中的第0个图层为可查询状态, 然后创建查询参数对象实例并给其相关参数赋值, 查询参数对象的查询类型属性设为 ESelectionType.Condition, 查询条件属性设为 "id>10"。最后调用 VectorMapDoc 的 select 方法实现属性条件查询。

**注意:** 为了方便展示查询结果, 本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以显示查询结果。

切换到源码显示窗口, 显示如下:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:ns1="zdims.map.*" fontSize="12"
xmlns:ns2="zdims.control.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" imsmap="{map}">
        </ns2:MapDocDataViewer>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="查询第0个图层中ID>10的要素"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;
            import Mapgis7.Webservice.BasLib.EnumLayerStatus;
            import Mapgis7.Webservice.BasLib.CMapSelectParam;
            import Mapgis7.Webservice.BasLib.ESelectionType;
            import Mapgis7.Webservice.BasLib.CWebSelectParam;

            public function callBack(e:Event):void
            {
                this.map.activeMapDoc=this.mapdoc;
                dataViewer.showResult(this.mapdoc.onSelect(e));
            }
        ]]>
    </mx:Script>
</mx:Application>
```

```

    }

    public function onClick():void
    {

        if(this.mapdoc.getMapLayerInfo(0).LayerStatus!=EnumLayerStatus.Selectable)
        {

            this.mapdoc.getMapLayerInfo(0).LayerStatus=EnumLayerStatus.Selectable;

            this.mapdoc.updateAllLayerInfo();
            Alert.show("图层状态设置成功,");
        }
        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.Condition;
        obj.WhereClause="id>10";
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param, callBack);
    }
}]>
</mx:Script>

</mx:Application>

```

## 第16问 怎样实现点击+条件查询

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第1个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByHitting，同时监听地图的单击事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedDot 方法得到鼠标选取的点，然后创建查询参数对象实例并给其相关参数赋值，把查询参数对象的查询条件属性 WhereClause 设为 "ID>0"，最后调用 VectorMapDoc 的 select 方法实现点击条件查询。

**注意：**为了方便展示查询结果，本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以显示查询结果。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMap id="map"

```

```

        x="106"
        y="95"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
<ns1:VectorMapDoc id="mapdoc"
        mapDocName="wh.map"
        serverAddress="127.0.0.1:5141">
</ns1:VectorMapDoc>
<ns2:MapDocDataViewer id="dataViewer" imsmap="{map}">
</ns2:MapDocDataViewer>
</ns1:IMSMMap>
<mx:Button x="184" y="31" label="点击查询第1个图层中ID>0的要素"
click="onClick()" />
<mx:Script>
    <![CDATA[
        import zdims.event.IMSMMapEvent;
        import zdims.util.IMSOperType;
        import Mapgis7.WebService.BasLib.EnumLayerStatus;
        import Mapgis7.WebService.BasLib.CMapSelectParam;
        import mx.controls.Alert;
        import Mapgis7.WebService.BasLib.EWebGeomType;
        import Mapgis7.WebService.BasLib.ESelectionType;
        import Mapgis7.WebService.BasLib.CWebSelectParam;
        import zdims.event.IMSMMapMouseEvent;
        public function init():void
        {

            this.map.addEventListener(IMSMMapEvent.MAP_OPERTYPECHANGE,onOperCh
ange);
        }
        public function onOperChange(e:Event):void
        {

            this.map.removeEventListener(IMSMMapMouseEvent.MAP_CLICK,onMouseCl
ick);
        }
        public function onMouseClick(e:Event):void
        {
            var obj:CWebSelectParam=new CWebSelectParam();
            obj.SelectionType=ESelectionType.Both;

```

```

        obj.WhereClause="ID>0";
        obj.GeomType=EWebGeomType.Point;
        obj.Geometry=this.map.graphicCanvas.getSelectedDot();
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param,callback);
    }
    public function callBack(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        dataViewer.showResult(this.mapdoc.onSelect(e));
    }
    public function onClick():void
    {
        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Selectable;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByHitting);

        this.map.addEventListener(IMSMapMouseEvent.MAP_CLICK,onMouseClicked);
    }
}]]>
</mx:Script>

</mx:Application>

```

## 第17问 怎样实现线+条件查询

参照第二章第3问实现地图文档显示, 添加一个 Button 控件, 在点击该 Button 时, 设置地图文档中的第1个图层为可查询状态, 然后设置地图操作类型为 IMSOperType.SelectByLine, 同时监听地图的双击事件, 在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedLine 方法得到鼠标选取的线, 然后创建查询参数对象实例并给其相关参数赋值, 把查询参数对象的查询条件属性 WhereClause 设为 "ID>0", 最后调用 VectorMapDoc 的 select 方法实现线条件查询

**注意:** 为了方便展示查询结果, 本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以显示查询结果。

切换到源码显示窗口, 显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"

```

```

fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSTMap id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" imsmmap="{map}">
        </ns2:MapDocDataViewer>
    </ns1:IMSTMap>
    <mx:Button x="76" y="10" label="线查询第1个图层中的ID>0的要素"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.IWebGeometry;
            import zdims.event.IMSTMapEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import Mapgis7.WebService.BasLib.EWebGeomType;
            import Mapgis7.WebService.BasLib.ESelectionType;
            import Mapgis7.WebService.BasLib.CWebSelectParam;
            import zdims.event.IMSTMapMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSTMapEvent.MAP_OPERTYPECHANGE,onOperChange);

            }
            public function onOperChange(e:Event):void
            {

                this.map.removeEventListener(IMSTMapMouseEvent.MAP_DOUBLECLICK,mouseCallBack);

            }
            public function mouseCallBack(e:Event):void
            {

```

```

        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.Both;
        obj.Geometry=this.map.graphicCanvas.getSelectedLine();
        if(obj.Geometry==null)
            return;
        obj.GeomType=EWebGeomType.Line;
        obj.NearDistance=0.0001;
        obj.WhereClause="id>0";
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param,callback);
    }
    public function callBack(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        dataViewer.showResult(this.mapdoc.onSelect(e));
    }
    public function onClick():void
    {
        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Selectable;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByLine);

        this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,mouseCallback);
    }
}]]>
</mx:Script>

</mx:Application>

```

## 第18问 怎样实现矩形+条件查询

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 1 个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByRect，同时监听地图的鼠标弹起事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedRect 方法得到鼠标选取的矩形，然后创建查询参数对象实例并给其相关参数赋值，把查询参数对象的查询条件属性 WhereClause 设为"ID>0"，最后调用 VectorMapDoc 的 select 方法实现矩形条件查询

**注意：**为了方便展示查询结果，本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以



显示查询结果。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMMap id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" imsmmap="{map}">
        </ns2:MapDocDataViewer>
    </ns1:IMSMMap>
    <mx:Button x="76" y="10" label="矩形查询第1个图层中ID>0的要素"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.IWebGeometry;
            import zdims.event.IMSMMapEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import mx.controls.Alert;
            import Mapgis7.WebService.BasLib.EWebGeomType;
            import Mapgis7.WebService.BasLib.ESelectionType;
            import Mapgis7.WebService.BasLib.CWebSelectParam;
            import zdims.event.IMSMMapMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSMMapEvent.MAP_OPERTYPECHANGE,onOperCh
ange);

            }
            public function onOperChange(e:Event):void
            {
```

```

        this.map.removeEventListener(IMSMapMouseEvent.MAP_MOUSEUP, mouseCa
llBack);
    }
    public function mouseCallBack(e:Event):void
    {
        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.Both;
        obj.Geometry=this.map.graphicCanvas.getSelectedRect();
        if(obj.Geometry==null)
            return;
        obj.WhereClause="id>0";
        obj.GeomType=EWebGeomType.Rect;
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param, callBack);
    }
    public function callBack(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        dataViewer.showResult(this.mapdoc.onSelect(e));
    }
    public function onClick():void
    {

        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Select
able;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByRect);

        this.map.addEventListener(IMSMapMouseEvent.MAP_MOUSEUP, mouseCallB
ack);
    }
    ]]>
</mx:Script>

</mx:Application>

```

## 第19问 怎样实现圆+条件查询

参照第二章第 3 问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 1 个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByCircle，同时监听地图的鼠标弹起事件，在事件处理函数中通过调用 IMSMap 的

graphicCanvas 的 getSelectedCircle 方法得到鼠标选取的圆，然后创建查询参数对象实例并给其相关参数赋值，把查询参数对象的查询条件属性 WhereClause 设为 "ID>10"，最后调用 VectorMapDoc 的 select 方法实现圆条件查询

**注意：**为了方便展示查询结果，本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以显示查询结果。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" imsmap="{map}">
        </ns2:MapDocDataViewer>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="画圆查询第1个图层中ID>10的要素"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.IWebGeometry;
            import zdims.event.IMSMAPEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import mx.controls.Alert;
            import Mapgis7.WebService.BasLib.EWebGeomType;
            import Mapgis7.WebService.BasLib.ESelectionType;
            import Mapgis7.WebService.BasLib.CWebSelectParam;
            import zdims.event.IMSMAPMouseEvent;
            public function init():void
            {
```

```

        this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE, onOperChange);
    }
    public function onOperChange(e:Event):void
    {

        this.map.removeEventListener(IMSMapMouseEvent.MAP_MOUSEUP, mouseCallback);
    }
    public function mouseCallback(e:Event):void
    {
        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.Both;

        obj.Geometry=this.map.graphicCanvas.getSelectedCircle();
        if(obj.Geometry==null)
        {
            return;
        }
        obj.WhereClause="id>10";
        obj.GeomType=EWebGeomType.Circle;
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param, callback);
    }
    public function callback(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        dataViewer.showResult(this.mapdoc.onSelect(e));
    }
    public function onClick():void
    {

        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Selectable;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByCircle);

        this.map.addEventListener(IMSMapMouseEvent.MAP_MOUSEUP, mouseCallback);
    }
]]>
</mx:Script>

</mx:Application>

```

## 第20问 怎样实现多边形+条件查询

参照第二章第 3 问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 1 个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByPolygon，同时监听地图的鼠标双击事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedPolygon 方法得到鼠标选取的多边形，然后创建查询参数对象实例并给其相关参数赋值，把查询参数对象的查询条件属性 WhereClause 设为"ID>10"，最后调用 VectorMapDoc 的 select 方法实现多边形条件查询

**注意：**为了方便展示查询结果，本示例使用了地图文档查询结果显示控件 MapDocDataViewer。在查询的回调函数中调用了 MapDocDataViewer 的 showResult 函数以显示查询结果。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMap id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" imsmap="{map}">
        </ns2:MapDocDataViewer>
    </ns1:IMSMap>
    <mx:Button x="76" y="10" label="多边形查询第1个图层中ID>10的要素"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.IWebGeometry;
            import zdims.event.IMSMapEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import mx.controls.Alert;
            import Mapgis7.WebService.BasLib.EWebGeomType;
            import Mapgis7.WebService.BasLib.ESelectionType;
```

```

import Mapgis7.WebService.BasLib.CWebSelectParam;
import zdims.event.IMSMapMouseEvent;
public function init():void
{

    this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE,onOperChange);

    public function onOperChange(e:Event):void
    {

        this.map.removeEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,mouseCallBack);

        public function mouseCallBack(e:Event):void
        {

            var obj:CWebSelectParam=new CWebSelectParam();
            obj.SelectionType=ESelectionType.Both;

            obj.Geometry=this.map.graphicCanvas.getSelectedPolygon();
            if(obj.Geometry==null)
            {
                return;
            }
            obj.WhereClause="id>10";
            obj.GeomType=EWebGeomType.Polygon;
            var param:CMapSelectParam=new CMapSelectParam();
            param.SelectParam=obj;
            this.mapdoc.select(param,callback);

        }
        public function callback(e:Event):void
        {

            this.map.activeMapDoc=this.mapdoc;
            dataViewer.showResult(this.mapdoc.onSelect(e));

        }
        public function onClick():void
        {

            this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Selectable;

            this.mapdoc.updateAllLayerInfo();
            this.map.setCurOper(IMSOperType.SelectByPolygon);

            this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,mouseCallBack);

        }
    }
}

```

```

]]>
</mx:Script>

</mx:Application>

```

## 第21问 怎样添加一个要素

参照第二章第3问实现地图文档显示, 为了方便的实现添加要素, 可以使用 Editor 控件来实现, 下面以添加多边形要素为例做示例。首先添加一个 Button 控件, 在点击该 Button 时, 设置地图文档中的第 0 个图层为可编辑状态, 然后设置地图操作类型为 IMSOperType.SelectByPolygon, 同时监听地图的鼠标双击事件, 事件处理函数为 Editor 控件的 addPolygon 函数。

**注意:** 该示例使用了 Editor 控件来快速实现添加要素的功能, 如果需要详细了解添加要素功能, 可查看 Editor 控件的源码。

切换到源码显示窗口, 显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
  <ns1:IMSTMap id="map"
    x="34"
    y="54"
    width="703"
    height="458"
    xMinMap="114.125602229914"
    xMaxMap="114.500788705197"
    yMinMap="30.4539323507469"
    yMaxMap="30.8291188260302">
    <ns1:VectorMapDoc id="mapdoc"
      mapDocName="wh.map"
      serverAddress="127.0.0.1:5141">
    </ns1:VectorMapDoc>
    <ns2:Editor id="editCtrl" imsmap="{map}">
    </ns2:Editor>
  </ns1:IMSTMap>
  <mx:Button x="76" y="10" label="在第0个图层上添加区要素"
click="onClick()" />
  <mx:Script>
    <![CDATA[
      import zdims.event.IMSTMapEvent;
      import zdims.util.IMSOperType;
      import Mapgis7.WebService.BasLib.EnumLayerStatus;
      import zdims.event.IMSTMapMouseEvent;

```

```

        public function init():void
        {

            this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE,onOperChange);

        }

        public function onOperChange(e:Event):void
        {

            this.map.removeEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,this.e
s.editCtrl.addPolygon);

        }

        public function onClick():void
        {

            this.mapdoc.getMapLayerInfo(0).LayerStatus=EnumLayerStatus.Editab
le;

            this.map.activeMapDoc=this.mapdoc;
            this.mapdoc.activeLayerIndex=0;
            this.mapdoc.updateAllLayerInfo();
            this.map.setCurOper(IMSOperType.SelectByPolygon);

            this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,this.e
ditCtrl.addPolygon);

        }

    ]]>
</mx:Script>

</mx:Application>

```

## 第22问 怎样编辑一个要素

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 1 个图层为可编辑状态，然后设置地图操作类型为 IMSOperType.SelectByPolygon，同时监听地图的双击事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedPolygon 方法得到鼠标选取的多线形对象，然后创建查询参数对象实例并给其相关参数赋值，最后调用 VectorMapDoc 的 select 方法实现多边形查询。在查询结果显示控件中可以通过右键菜单中的编辑功能打开编辑控件对要素进行编辑。

**注意：**为了快速实现编辑要素功能，本示例使用了地图文档查询结果显示控件 MapDocDataViewer 和编辑控件 Editor，如果需要详细了解编辑要素功能，可查看 MapDocDataViewer 控件和 Editor 控件的源码。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
```



```

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSTMap id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" editor="{editCtrl}"
imsmap="{map}">
        </ns2:MapDocDataViewer>
        <ns2:Editor id="editCtrl" imsmap="{map}">
        </ns2:Editor>
    </ns1:IMSTMap>
    <mx:Button x="76" y="10" label="多边形查询第1个图层中ID>10的要素"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import zdims.event.IMSTMapEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import mx.controls.Alert;
            import Mapgis7.WebService.BasLib.ESelectionType;
            import Mapgis7.WebService.BasLib.CWebSelectParam;
            import zdims.event.IMSTMapMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSTMapEvent.MAP_OPERTYPECHANGE,onOperCh
ange);

            }
            public function onOperChange(e:Event):void
            {

                this.map.removeEventListener(IMSTMapMouseEvent.MAP_DOUBLECLICK,mou

```

```

seCallBack);
    }
    public function mouseCallBack(e:Event):void
    {
        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.Both;

        obj.Geometry=this.map.graphicCanvas.getSelectedPolygon();
        if(obj.Geometry==null)
            return;
        obj.WhereClause="id>10";
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param,callBack);
    }
    public function callBack(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        this.map.activeMapDoc.activeLayerIndex=1;
        dataViewer.showResult(this.mapdoc.onSelect(e));
        Alert.show("查询成功，您可以通过右键菜单来编辑选择的记录。");
    }
    public function onClick():void
    {
        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.
Editable;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByPolygon);

        this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK,mouseC
allBack);
    }
]]>
</mx:Script>

</mx:Application>

```

## 第23问 怎样删除一个要素

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第1个图层为可编辑状态，然后设置地图操作类型为 IMSOperType.SelectByPolygon，同时监听地图的双击事件，在事件处理函数中通过调用 IMSMap 的

graphicCanvas 的 getSelectedPolygon 方法得到鼠标选取的多线形对象，然后创建查询参数对象实例并给其相关参数赋值，最后调用 VectorMapDoc 的 select 方法实现多边形查询。在查询结果显示控件中可以通过右键菜单中的删除功能来删除要素。

**注意：**为了快速实现删除要素功能，本示例使用了地图文档查询结果显示控件 MapDocDataViewer 和编辑控件 Editor，如果需要详细了解删除要素功能，可查看 MapDocDataViewer 控件和 Editor 控件的源码。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:MapDocDataViewer id="dataViewer" editor="{editCtrl}"
imsmap="{map}">
        </ns2:MapDocDataViewer>
        <ns2:Editor id="editCtrl" imsmap="{map}">
        </ns2:Editor>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="多边形查询第1个图层中ID>10的要素"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import zdims.event.IMSMAPEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import mx.controls.Alert;
            import Mapgis7.WebService.BasLib.ESelectionType;
            import Mapgis7.WebService.BasLib.CWebSelectParam;
            import zdims.event.IMSMAPMouseEvent;
            public function init():void
            {
```

```

        this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE, onOperChange);
    }
    public function onOperChange(e:Event):void
    {

        this.map.removeEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK, mouseCallback);
    }
    public function mouseCallback(e:Event):void
    {
        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.Both;

        obj.Geometry=this.map.graphicCanvas.getSelectedPolygon();
        if(obj.Geometry==null)
        {
            return;
        }
        obj.WhereClause="id>10";
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param, callback);
    }
    public function callback(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        this.map.activeMapDoc.activeLayerIndex=1;
        dataViewer.showResult(this.mapdoc.onSelect(e));
        Alert.show("查询成功，您可以通过右键菜单来删除选择的记录。");
    }
    public function onClick():void
    {

        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.
        Editable;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByPolygon);

        this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK, mouseCallback);
    }
    ]]>
</mx:Script>
</mx:Application>

```

## 第24问 怎样对一个要素做缓冲区分析

参照第二章第3问实现地图文档显示, 添加一个 Button 控件, 在点击该 Button 时, 设置地图文档中的第1个图层为可查询状态, 然后设置地图操作类型为 IMSOperType.SelectByHitting, 同时监听地图的单击事件, 在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedDot 方法得到鼠标选取的点, 然后创建查询参数对象实例并给其相关参数赋值, 最后调用 VectorMapDoc 的 select 方法实现点击查询。通过 VectorMapDoc 的 onSelect 可以得到查询结果集, 创建一个 CGetObjByID 并从得到的结果集中获取第1个图层结果记录的第一条记录的要素 ID, 给 CGetObjByID 对象的 LayerIndex 属性赋值为 1。最后调用缓冲区分析控件 BufferAnalyse 的 setTargetFeature 函数设置要进行缓冲区分析的目的要素, 再通过点击缓冲区分析控件的提交按钮即可对要素进行缓冲区分析。

**注意:** 为了快速实现缓冲区分析和结果显示功能, 本示例使用了缓冲区分析控件 BufferAnalyse 和图层查询结果显示控件 LayerDataViewer。如果想深入了解缓冲区分析功能, 请查看缓冲区分析控件 BufferAnalyse 的源码。

切换到源码显示窗口, 显示如下:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMap id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:BufferAnalyse id="bufferCtrl"
layerDataViewer="{layerData}" imsmap="{map}">
        </ns2:BufferAnalyse>
        <layer:LayerDataViewer id="layerData" imsmap="{map}">
        </layer:LayerDataViewer>
    </ns1:IMSMap>
    <mx:Button x="76" y="10" label="点击查询第1个图层中的要素进行缓冲区分析"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.CAttDataRow;
```

```

import Mapgis7.WebService.BasLib.CAttDataTable;
import Mapgis7.WebService.BasLib.CAttDataSet;
import Mapgis7.WebService.BasLib.CGetObjByID;
import Mapgis7.WebService.BasLib.CMapSelectAndGetAtt;
import zdims.event.IMSMapEvent;
import zdims.util.IMSOperType;
import Mapgis7.WebService.BasLib.EnumLayerStatus;
import Mapgis7.WebService.BasLib.CMapSelectParam;
import mx.controls.Alert;
import Mapgis7.WebService.BasLib.EWebGeomType;
import Mapgis7.WebService.BasLib.ESelectionType;
import Mapgis7.WebService.BasLib.CWebSelectParam;
import zdims.event.IMSMapMouseEvent;
public function init():void
{

    this.map.addListener(IMSMapEvent.MAP_OPERTYPECHANGE,onOperChange);

    public function onOperChange(e:Event):void
    {

        this.map.removeListener(IMSMapMouseEvent.MAP_CLICK,onMouseClicked);

        public function onMouseClick(e:Event):void
        {
            var obj:CWebSelectParam=new CWebSelectParam();
            obj.SelectionType=ESelectionType.SpatialRange;
            obj.GeomType=EWebGeomType.Point;
            obj.Geometry=this.map.graphicCanvas.getSelectedDot();
            var param:CMapSelectParam=new CMapSelectParam();
            param.SelectParam=obj;
            this.mapdoc.select(param,callback);
        }
        public function callback(e:Event):void
        {
            this.map.activeMapDoc=this.mapdoc;
            this.map.activeMapDoc.activeLayerIndex=1;
            var result:CMapSelectAndGetAtt=this.mapdoc.onSelect(e);
            if(result==null||result.Count[0][1]==0)
            {
                Alert.show("未查到要素，请重新点击查询");
                return;
            }
        }
    }
}

```

```

    }
    else
        Alert.show("已查询到目的要素，现在可以点击缓冲区分析控件中的
提交按钮进行缓冲区分析！");
        var targetObj:CGetObjByID=new CGetObjByID();
        targetObj.FeatureID=((result.AttDS[0] as
CAttDataSet).attTables[1] as CAttDataTable).Rows[0] as
CAttDataRow).FID;
        targetObj.LayerIndex=1;
        this.bufferCtrl.setTargetFeature(targetObj);
        this.bufferCtrl.visible=true;
    }
    public function onClick():void
    {

        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Select
able;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByHitting);

        this.map.addEventListener(IMSMapMouseEvent.MAP_CLICK,onMouseClick
);
    }
}]>
</mx:Script>

</mx:Application>

```

## 第25问 怎样获取要素空间坐标信息并在地图上闪烁

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第1个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByHitting，同时监听地图的单击事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedDot 方法得到鼠标选取的点，然后创建查询参数对象实例并给其相关参数赋值，最后调用 VectorMapDoc 的 select 方法实现点击查询。通过 VectorMapDoc 的 onSelect 可以得到查询结果集，创建一个 CGetObjByID 并从得到的结果集中获取第1个图层结果记录的第一条记录的要素 ID，给 CGetObjByID 对象的 LayerIndex 属性赋值为 1。通过调用 VectorMapDoc 的 getFeatureByID 函数获取要素信息，在 getFeatureByID 的回调函数 onGetFeature 中调用 VectorMapDoc 的 onGetFeatureByID 函数得到服务器返回的要素信息对象，然后调用 IMSMap 的 graphicCanvas 属性的 flashGeo 函数闪烁要素。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"

```

```

creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="点击查询第1个图层中的要素"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.SFeature;
            import Mapgis7.WebService.BasLib.CAttDataRow;
            import Mapgis7.WebService.BasLib.CAttDataTable;
            import Mapgis7.WebService.BasLib.CAttDataSet;
            import Mapgis7.WebService.BasLib.CGetObjByID;
            import Mapgis7.WebService.BasLib.CMapSelectAndGetAtt;
            import zdims.event.IMSMAPEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import mx.controls.Alert;
            import Mapgis7.WebService.BasLib.EWebGeomType;
            import Mapgis7.WebService.BasLib.ESelectionType;
            import Mapgis7.WebService.BasLib.CWebSelectParam;
            import zdims.event.IMSMAPMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSMAPEvent.MAP_OPERTYPECHANGE,onOperCh
ange);

            }
            public function onOperChange(e:Event):void
            {

```



```

        this.map.removeListener(IMSMapMouseEvent.MAP_CLICK, onMouseClick);
    }
    public function onMouseClick(e:Event):void
    {
        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.SpatialRange;
        obj.GeomType=EWebGeomType.Point;
        obj.Geometry=this.map.graphicCanvas.getSelectedDot();
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param, callBack);
    }
    public function callBack(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        this.map.activeMapDoc.activeLayerIndex=1;
        var result:CMapSelectAndGetAtt=this.mapdoc.onSelect(e);
        if(result==null || result.Count[0][1]==0)
        {
            Alert.show("未查到要素，请重新点击查询");
            return;
        }
        var targetObj:CGetObjByID=new CGetObjByID();
        targetObj.FeatureID=((result.AttDS[0] as
CAttDataSet).attTables[1] as CAttDataTable).Rows[0] as
CAttDataRow).FID;
        targetObj.LayerIndex=1;
        this.mapdoc.getFeatureByID(targetObj, onGetFeature);
    }
    public function onGetFeature(e:Event):void
    {
        var sf:SFeature=this.mapdoc.onGetFeatureByID(e);
        this.map.graphicCanvas.flashGeo(sf.fGeom);
    }
    public function onClick():void
    {
        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Selectable;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByHitting);
    }

```

```

        this.map.addEventListener(IMSMapMouseEvent.MAP_CLICK, onMouseClick);
    }
]]>
</mx:Script>

</mx:Application>

```

## 第26问 怎样获取要素图形参数

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第1个图层为可查询状态，然后设置地图操作类型为 IMSOperType.SelectByHitting，同时监听地图的单击事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedDot 方法得到鼠标选取的点，然后创建查询参数对象实例并给其相关参数赋值，最后调用 VectorMapDoc 的 select 方法实现点击查询。通过 VectorMapDoc 的 onSelect 可以得到查询结果集，创建一个 CGetObjByID 并从得到的结果集中获取第1个图层结果记录的第一条记录的要素 ID，给 CGetObjByID 对象的 LayerIndex 属性赋值为 1。通过调用 VectorMapDoc 的 getFeatureStyleInfo 函数获取要素图形样式，在 getFeatureStyleInfo 的回调函数 onGetFeature 中调用 VectorMapDoc 的 onGetFeatureStyleInfo 函数得到服务器返回的要素图形参数。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMap id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMap>
    <mx:Button x="76" y="10" label="点击查询第1个图层中的要素"
click="onClick()" />
</mx:Script>

```

```

<![CDATA[
    import Mapgis7.WebService.BasLib.WebGraphicsInfo;
    import Mapgis7.WebService.BasLib.SFeature;
    import Mapgis7.WebService.BasLib.CAttDataRow;
    import Mapgis7.WebService.BasLib.CAttDataTable;
    import Mapgis7.WebService.BasLib.CAttDataSet;
    import Mapgis7.WebService.BasLib.CGetObjByID;
    import Mapgis7.WebService.BasLib.CMapSelectAndGetAtt;
    import zdims.event.IMSMapEvent;
    import zdims.util.IMSOperType;
    import Mapgis7.WebService.BasLib.EnumLayerStatus;
    import Mapgis7.WebService.BasLib.CMapSelectParam;
    import mx.controls.Alert;
    import Mapgis7.WebService.BasLib.EWebGeomType;
    import Mapgis7.WebService.BasLib.ESelectionType;
    import Mapgis7.WebService.BasLib.CWebSelectParam;
    import zdims.event.IMSMapMouseEvent;
    public function init():void
    {

        this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE,onOperChange);

    }
    public function onOperChange(e:Event):void
    {

        this.map.removeEventListener(IMSMapMouseEvent.MAP_CLICK,onMouseClicked);

    }
    public function onMouseClick(e:Event):void
    {
        var obj:CWebSelectParam=new CWebSelectParam();
        obj.SelectionType=ESelectionType.SpatialRange;
        obj.GeomType=EWebGeomType.Point;
        obj.Geometry=this.map.graphicCanvas.getSelectedDot();
        var param:CMapSelectParam=new CMapSelectParam();
        param.SelectParam=obj;
        this.mapdoc.select(param,callback);
    }
    public function callback(e:Event):void
    {
        this.map.activeMapDoc=this.mapdoc;
        this.map.activeMapDoc.activeLayerIndex=1;
        var result:CMapSelectAndGetAtt=this.mapdoc.onSelect(e);
    }
}

```

```

        if (result == null || result.Count[0][1] == 0)
        {
            Alert.show("未查到要素，请重新点击查询");
            return;
        }
        var targetObj: CGetObjByID = new CGetObjByID();
        targetObj.FeatureID = ((result.AttDS[0] as
CAttDataSet).attTables[1] as CAttDataTable).Rows[0] as
CAttDataRow).FID;
        targetObj.LayerIndex = 1;

        this.mapdoc.getFeatureStyleInfo(targetObj, onGetFeature);
    }
    public function onGetFeature(e: Event): void
    {
        var
obj: WebGraphicsInfo = this.mapdoc.onGetFeatureStyleInfo(e);
        Alert.show("选中的区要素的填充颜色值
=" + obj.RegInfo.FillColor);
    }
    public function onClick(): void
    {
        this.mapdoc.getMapLayerInfo(1).LayerStatus = EnumLayerStatus.Select
able;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByHitting);

        this.map.addEventListener(IMSMapMouseEvent.MAP_CLICK, onMouseClick
);
    }
]]>
</mx:Script>

</mx:Application>

```

## 第27问 怎样更新要素图形参数

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第1个图层为可编辑状态，然后设置地图操作类型为 IMSOperType.SelectByHitting，同时监听地图的单击事件，在事件处理函数中通过调用 IMSMap 的 graphicCanvas 的 getSelectedDot 方法得到鼠标选取的点，然后创建查询参数对象实例并给其相关参数赋值，最后调用 VectorMapDoc 的 select 方法实现点击查询。通过

VectorMapDoc 的 onSelect 可以得到查询结果集, 创建一个 CGetObjByID 并从得到的结果集中获取第 1 个图层结果记录的第一条记录的要素 ID, 给 CGetObjByID 对象的 LayerIndex 属性赋值为 1. 通过调用 VectorMapDoc 的 getFeatureStyleInfo 函数获取要素图形样式, 在 getFeatureStyleInfo 的回调函数 onGetFeature 中调用 VectorMapDoc 的 onGetFeatureStyleInfo 函数得到服务器返回的要素图形参数。本示例代码将得到的区要素的填充颜色值设置为 1, 然后调用 VectorMapDoc 的 updateFeature 函数更新选中的要素的图形参数。

切换到源码显示窗口, 显示如下:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="点击查询第1个图层中的要素"
click="onClick()" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.EsfclsGeomType;
            import Mapgis7.WebService.BasLib.CMapFeatureInfo;
            import Mapgis7.WebService.BasLib.WebGraphicsInfo;
            import Mapgis7.WebService.BasLib.SFeature;
            import Mapgis7.WebService.BasLib.CAttDataRow;
            import Mapgis7.WebService.BasLib.CAttDataTable;
            import Mapgis7.WebService.BasLib.CAttDataSet;
            import Mapgis7.WebService.BasLib.CGetObjByID;
            import Mapgis7.WebService.BasLib.CMapSelectAndGetAtt;
            import zdims.event.IMSMAPEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import Mapgis7.WebService.BasLib.CMapSelectParam;
            import mx.controls.Alert;
```

```

import Mapgis7.WebService.BasLib.EWebGeomType;
import Mapgis7.WebService.BasLib.ESelectionType;
import Mapgis7.WebService.BasLib.CWebSelectParam;
import zdims.event.IMSMapMouseEvent;
public function init():void
{

    this.map.addEventListener(IMSMapEvent.MAP_OPERTYPECHANGE,onOperChange);

    public function onOperChange(e:Event):void
    {

        this.map.removeEventListener(IMSMapMouseEvent.MAP_CLICK,onMouseClicked);

        public function onMouseClick(e:Event):void
        {
            var obj:CWebSelectParam=new CWebSelectParam();
            obj.SelectionType=ESelectionType.SpatialRange;
            obj.GeomType=EWebGeomType.Point;
            obj.Geometry=this.map.graphicCanvas.getSelectedDot();
            var param:CMapSelectParam=new CMapSelectParam();
            param.SelectParam=obj;
            this.mapdoc.select(param,callback);
        }
        public function callback(e:Event):void
        {
            this.map.activeMapDoc=this.mapdoc;
            this.map.activeMapDoc.activeLayerIndex=1;
            var result:CMapSelectAndGetAtt=this.mapdoc.onSelect(e);
            if(result==null||result.Count[0][1]==0)
            {
                Alert.show("未查到要素，请重新点击查询");
                return;
            }
            var targetObj:CGetObjByID=new CGetObjByID();
            targetObj.FeatureID=((result.AttDS[0] as
CAttDataSet).attTables[1] as CAttDataTable).Rows[0] as
CAttDataRow).FID;
            targetObj.LayerIndex=1;

            this.mapdoc.getFeatureStyleInfo(targetObj,function(e:Event):void{
onGetFeature(e,targetObj.FeatureID)});

```

```

    }
    public function onGetFeature(e:Event,fid:int):void
    {
        var
obj:WebGraphicsInfo=this.mapdoc.onGetFeatureStyleInfo(e);
        Alert.show("选中的区要素的填充颜色值
="+obj.RegInfo.FillColor);
        var finfo:CMapFeatureInfo=new CMapFeatureInfo();
        finfo.LayerIndex=1;
        finfo.FSet=new SFeature();
        finfo.FSet.ftype=ESFclsGeomType.Reg;
        finfo.FSet.FID=fid;
        obj.RegInfo.FillColor=1;
        finfo.GInfo=obj;
        this.mapdoc.updateFeature(finfo,onUpdateFeature);
    }
    public function onUpdateFeature(e:Event):void
    {
        if(this.mapdoc.onUpdateFeature(e).OperResult)
            Alert.show("选中的要素的填充颜色值已更新为1");
    }
    public function onClick():void
    {
        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Editab
le;

        this.mapdoc.updateAllLayerInfo();
        this.map.setCurOper(IMSOperType.SelectByHitting);

        this.map.addEventListener(IMSMapMouseEvent.MAP_CLICK,onMouseClicked
);
    }
]]>
</mx:Script>

</mx:Application>

```

## 第28问 怎样实现圆裁剪

参照第二章第 3 问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 0 个图层为可编辑状态，然后设置地图操作类型为 IMSOperType.SelectByCircle，同时监听地图的鼠标弹起事件，事件处理函数这里设置为裁剪分析控件 ClipAnalyse 的 clipDocByCircle 函数。

注意：为了快速实现圆裁剪功能，本示例使用了裁剪分析控件 ClipAnalyse 和图层查询结果显示控件 LayerDataViewer。如果要深入了解裁剪分析的过程，请查看 ClipAnalyse 控件的源码。本示例演示的是对地图文档的某个图层进行裁剪，如果是对图层控件 VectorLayer 进行裁剪，则鼠标事件回调函数需要设置为 ClipAnalyse 的 clipLayerByCircle 函数。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMAP id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:ClipAnalyse id="clip" imsmap="{map}"
layerDataViewer="{layerData}">
        </ns2:ClipAnalyse>
        <layer:LayerDataViewer id="layerData" imsmap="{map}">
        </layer:LayerDataViewer>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="圆裁剪第1个图层" click="onClick()"/>
    <mx:Script>
        <![CDATA[
            import zdims.event.IMSMAPEvent;
            import zdims.util.IMSOperType;
            import Mapgis7.WebService.BasLib.EnumLayerStatus;
            import zdims.event.IMSMAPMouseEvent;
            public function init():void
            {

                this.map.addEventListener(IMSMAPEvent.MAP_OPERTYPECHANGE,onOperChange);

            }
            public function onOperChange(e:Event):void
            {
```



```

        this.map.removeEventListener(IMSMapMouseEvent.MAP_MOUSEUP, this.clipDocByCircle);
    }
    public function onClick():void
    {

        this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Editable;

        this.mapdoc.updateAllLayerInfo();
        this.map.activeMapDoc=this.mapdoc;
        this.map.activeMapDoc.activeLayerIndex=1;
        this.map.setCurOper(IMSOperType.SelectByCircle);

        this.map.addEventListener(IMSMapMouseEvent.MAP_MOUSEUP, this.clipDocByCircle);
    }
    ]]>
</mx:Script>

</mx:Application>

```

## 第29问 怎样实现多边形裁剪

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，设置地图文档中的第 0 个图层为可编辑状态，然后设置地图操作类型为 IMSOperType.SelectByPolygon，同时监听地图的鼠标的双击事件，事件处理函数这里设置为裁剪分析控件 ClipAnalyse 的 clipDocByPolygon 函数。

**注意：**为了快速实现多边形裁剪功能，本示例使用了裁剪分析控件 ClipAnalyse 和图层查询结果显示控件 LayerDataViewer。如果要深入了解裁剪分析的过程，请查看 ClipAnalyse 控件的源码。本示例演示的是对地图文档的某个图层进行裁剪，如果是对图层控件 VectorLayer 进行裁剪，则鼠标事件回调函数需要设置为 ClipAnalyse 的 clipLayerByPolygon 函数。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMap id="map"
        x="34"
        y="54"
        width="703"

```

```

        height="458"
        xminMap="114.125602229914"
        xmaxMap="114.500788705197"
        yminMap="30.4539323507469"
        ymaxMap="30.8291188260302">
<ns1:VectorMapDoc id="mapdoc"
        mapDocName="wh.map"
        serverAddress="127.0.0.1:5141">
</ns1:VectorMapDoc>
<ns2:ClipAnalyse id="clip" imsmap="{map}"
layerDataViewer="{layerData}">
</ns2:ClipAnalyse>
<layer:LayerDataViewer id="layerData" imsmap="{map}">
</layer:LayerDataViewer>
</ns1:IMSMAP>
<mx:Button x="76" y="10" label="多边形裁剪第1个图层"
click="onClick()" />
<mx:Script>
<![CDATA[
        import zdims.event.IMSMAPEvent;
        import zdims.util.IMSOperType;
        import Mapgis7.WebService.BasLib.EnumLayerStatus;
        import zdims.event.IMSMAPMouseEvent;
        public function init():void
        {

            this.map.addEventListener(IMSMAPEvent.MAP_OPERTYPECHANGE,onOperChange);

        }
        public function onOperChange(e:Event):void
        {

            this.map.removeEventListener(IMSMAPMouseEvent.MAP_DOUBLECLICK,this);
            s.clip.clipDocByPolygon();
        }
        public function onClick():void
        {

            this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Editab
le;

            this.mapdoc.updateAllLayerInfo();
            this.map.activeMapDoc=this.mapdoc;
            this.map.activeMapDoc.activeLayerIndex=1;
            this.map.setCurOper(IMSOperType.SelectByPolygon);

```

```

        this.map.addEventListener(IMSMapMouseEvent.MAP_DOUBLECLICK, this.clickDocByPolygon);
    }
]]>
</mx:Script>

</mx:Application>

```

## 第30问 怎样实现拓扑分析

参照第二章第 3 问实现地图文档显示, 添加一个 Button 控件, 在点击该 Button 时, 设置地图文档中的第 1 个图层为可查询状态, 然后即可在地图文档拓扑分析控件 TopAnalyse 中进行操作, 选取需要拓扑分析的两个要素, 然后点击拓扑分析控件中的提交按钮进行拓扑分析。

实现拓扑分析功能首先要实现地图要素的查询功能, 查询到要素后需要获取要素的空间信息, 然后才能对两个要素的空间信息进行拓扑分析。

**注意:** 为了快速实现地图文档要素拓扑分析功能, 本示例使用了拓扑分析控件 TopAnalyse。如果需要对 VectorLayer 控件中的图层进行拓扑分析, 需要使用图层拓扑分析控件 LayerTopAnalyse。若要深入了解拓扑分析功能, 请查看 TopAnalyse 和 LayerTopAnalyse 控件的源码。

切换到源码显示窗口, 显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMap id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:TopAnalyse id="topCtrl" imsmap="{map}" >
        </ns2:TopAnalyse>
    </ns1:IMSMap>

```

```

        <mx:Button x="76" y="10" label="设置第1个图层为可查状态"
click="onClick()" />
        <mx:Script>
            <![CDATA[
                import Mapgis7.WebService.BasLib.EnumLayerStatus;
                public function onClick():void
                {

                    this.mapdoc.getMapLayerInfo(1).LayerStatus=EnumLayerStatus.Select
able;

                    this.mapdoc.updateAllLayerInfo();
                    this.map.activeMapDoc=this.mapdoc;
                    this.map.activeMapDoc.activeLayerIndex=1;
                    this.topCtrl.visible=true;
                }
            ]]>
        </mx:Script>

    </mx:Application>

```

## 第31问 怎样实现路径分析

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，显示路径分析控件。本示例使用的网络类是 MapGisLocal 下面的 world 数据库下面的“武汉道路网”，示例代码中需要给路径分析控件的相关属性赋值（GDBSvrName="mapgislocal" GDBName="world" NetLayerName="武汉道路网" vectorObj="{mapdoc}" imsmap="{map}"）才能正常的使用路径分析功能。

**注意：**为了快速实现路径分析功能，本示例使用了路径分析控件 NetAnalyse。若要深入了解路径分析功能，请查看 NetAnalyse 控件的源码。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMMap id="map"
        x="34"
        y="54"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"

```

```

        yMaxMap="30.8291188260302">
<ns1:VectorMapDoc id="mapdoc"
        mapDocName="wh.map"
        serverAddress="127.0.0.1:5141">
</ns1:VectorMapDoc>
<ns2:NetAnalyse id="netAnalyse" GDBSvrName="mapgislocal"
GDBName="world" NetLayerName="武汉道路网" vectorObj="{mapdoc}"
imsmap="{map}" >
</ns2:NetAnalyse>
</ns1:IMSMap>
<mx:Button x="76" y="10" label="显示网络分析控件" click="onClick()" />
<mx:Script>
<![CDATA[
        public function onClick():void
        {
            this.netAnalyse.visible=true;
        }
    ]]>
</mx:Script>

</mx:Application>

```

## 第32问 怎样获取 MapGIS 数据源列表

参照第二章第 3 问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，创建一个分析类 SpacialAnalyse 然后调用其 getGdbServerList 函数获取数据源列表。

**注意：**分析类 SpacialAnalyse 的构造函数中的参数必需传入矢量图层对象，它可以是 VectorMapDoc 或 VectorLayer。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMap id="map"
        x="34"
        y="72"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">

```

```

        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMap>
    <mx:Button x="76" y="10" label="获取数据源" click="onClick()" />
    <mx:TextArea id="datalist" x="175" y="0" width="538" height="64" />
    <mx:Script>
        <![CDATA[
            import zdims.util.SpatialAnalyse;
            private var spatialObj:SpatialAnalyse=null;
            public function onClick():void
            {
                if(spatialObj==null)
                    spatialObj=new SpatialAnalyse(mapdoc);
                spatialObj.getGdbServerList(onGetData);
            }
            public function onGetData(e:Event):void
            {
                this.datalist.text=spatialObj.onGetGdbList(e).Name.join(",");
            }
        ]]>
    </mx:Script>
</mx:Application>

```

## 第33问 怎样获取 MapGIS 数据源中的数据库列表

参照第二章第 3 问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，创建一个分析类 SpatialAnalyse 然后调用其 getGdbList 函数获取数据库列表，本示例演示的是获取数据源 MapGisLocal 中的数据库列表，所以 CGetGdbList 中的数据源属性设为 MapGisLocal，用户名和密码都设为空。在实际开发过程中可根据实际情况赋值。

**注意：**分析类 SpatialAnalyse 的构造函数中的参数必需传入矢量图层对象，它可以是 VectorMapDoc 或 VectorLayer。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
    fontSize="12" xmlns:ns2="zdims.control.*"
    xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMap id="map"
        x="35"
        y="127"

```

```

        width="703"
        height="458"
        xminMap="114.125602229914"
        xmaxMap="114.500788705197"
        yminMap="30.4539323507469"
        ymaxMap="30.8291188260302">
<ns1:VectorMapDoc id="mapdoc"
        mapDocName="wh.map"
        serverAddress="127.0.0.1:5141">
</ns1:VectorMapDoc>
</ns1:IMSMap>
<mx:Button x="76" y="10" label="获取数据源MapGisLocal中的GDB列表"
click="onClick()" />
<mx:TextArea id="datalist" x="76" y="42" width="538" height="64"/>
<mx:Script>
    <![CDATA[
        import Mapgis7.WebService.BasLib.CGetGdbList;
        import zdims.util.SpatialAnalyse;
        private var spatialObj:SpatialAnalyse=null;
        public function onClick():void
        {
            if(spatialObj==null)
                spatialObj=new SpatialAnalyse(mapdoc);
            var obj:CGetGdbList=new CGetGdbList();
            obj.GdbSvrName="MapGisLocal";
            obj.GdbSvrUser="";
            obj.GdbSvrPwds="";
            spatialObj.getGdbList(obj,onGetData);
        }
        public function onGetData(e:Event):void
        {
            this.datalist.text=spatialObj.onGetGdbList(e).Name.join(",");
        }
    ]]>
</mx:Script>
</mx:Application>

```

## 第34问 怎样获取 MapGIS 数据库中的图层列表

参照第二章第 3 问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，创建一个分析类 SpatialAnalyse 然后调用其 getXClsList 函数获取数据库中某种类型的图层列表，本示例演示的是获取数据源 MapGisLocal 中的数据库 world 中的简单要素类列表，所以

CGetGdbList 中的数据源属性设为 MapGisLocal, 数据库名称设为 world, 图层类型设为简单要素类 EXClsType.SFeatureCls, 用户名和密码都设为空。在实际开发过程中可根据实际情况赋值。

**注意：**分析类 SpacialAnalyse 的构造函数中的参数必需传入矢量图层对象，它可以是 VectorMapDoc 或 VectorLayer。

切换到源码显示窗口，显示如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMap id="map"
        x="35"
        y="127"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMap>
    <mx:Button x="76" y="10" label="获取数据源MapGisLocal中的world数据库中的简单要素类图层" click="onClick()" />
    <mx:TextArea id="datalist" x="76" y="42" width="538" height="64" />
    <mx:Script>
        <![CDATA[
            import Mapgis7.WebService.BasLib.EXClsType;
            import Mapgis7.WebService.BasLib.CGetXClsList;
            import Mapgis7.WebService.BasLib.CGetGdbList;
            import zdims.util.SpacialAnalyse;
            private var spatialObj:SpacialAnalyse=null;
            public function onClick():void
            {
                if(spatialObj==null)
                    spatialObj=new SpacialAnalyse(mapdoc);
                var obj:CGetXClsList=new CGetXClsList();
                obj.GdbSvrName="MapGisLocal";
                obj.GdbName="world";
                obj.XClsType=EXClsType.SFeatureCls;
                obj.GdbSvrUser="";
            }
        ]]>
    </mx:Script>
</Application>
```



```

        obj.GdbSvrPwds="";
        spatialObj.getXClsList(obj,onGetData);
    }
    public function onGetData(e:Event):void
    {

        this.datalist.text=spatialObj.onGetXClsList(e).Name.join(",");
    }
    ]]>
</mx:Script>
</mx:Application>

```

## 第35问 怎样通过图层索引获取图层名称

参照第二章第3问实现地图文档显示,添加一个Button控件,在点击该Button时,通过调用地图文档对象的getLayerInfo函数获取地图文档中第0个图层的名称等信息。

**注意:** 本示例演示的是获取地图文档中的图层名称,如果使用VectorLayer控件,则可以通过其layerObj属性获取到其各个图层访问信息。

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMAP id="map"
        x="35"
        y="127"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="获取地图文档中第0个图层的名称"
click="onClick()" />
    <mx:TextArea id="datalist" x="76" y="42" width="538" height="64"/>
    <mx:Script>
        <![CDATA[
            public function onClick():void

```

```

        {

            this.datalist.text=this.mapdoc.getLayerInfo(0).LayerDataName;
        }
    ]]>
</mx:Script>
</mx:Application>

```

## 第36问 怎样实现两个图层叠加分析

参照第二章第3问实现地图文档显示，添加一个 Button 控件，在点击该 Button 时，显示叠加分析控件。

**注意：**为了快速实现叠加分析功能，本示例使用了叠加分析控件 OverLayAnalyse。若要深入了解拓扑分析功能，请查看 OverLayAnalyse 控件的源码。

切换到源码显示窗口，显示如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSMAP id="map"
        x="24"
        y="61"
        width="703"
        height="458"
        xminMap="114.125602229914"
        xmaxMap="114.500788705197"
        yminMap="30.4539323507469"
        ymaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
        <ns2:OverLayAnalyse id="overlayObj" imsmap="{map}"
vectorObj="{mapdoc}" layerDataViewer="{layerData}">
        </ns2:OverLayAnalyse>
        <layer:LayerDataViewer id="layerData" imsmap="{map}">
        </layer:LayerDataViewer>
    </ns1:IMSMAP>
    <mx:Button x="76" y="10" label="显示叠加分析控件" click="onClick()"/>
    <mx:Script>
        <![CDATA[
            public function onClick():void

```

```

        {
            this.overlayObj.visible=true;
        }
    ]]>
</mx:Script>
</mx:Application>

```

## 第37问 怎样在操作结束后把鼠标切换回空状态

参照第二章第 3 问实现地图文档显示, 添加两个 Button 控件, 分别响应 onClick 和 onClick1 两个函数。onClick 函数将鼠标操作切换为放大地图操作, onClick1 函数清空鼠标操作。

切换到源码显示窗口, 显示如下:

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:ns1="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*">
    <ns1:IMSTMap id="map"
        x="41"
        y="40"
        width="703"
        height="458"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <ns1:VectorMapDoc id="mapdoc"
            mapDocName="wh.map"
            serverAddress="127.0.0.1:5141">
        </ns1:VectorMapDoc>
    </ns1:IMSTMap>
    <mx:Button x="76" y="10" label="拉框放大地图操作" click="onClick()"/>
    <mx:Button x="244" y="10" label="清空操作" click="onClick1()"/>
    <mx:Script>
        <![CDATA[
            import zdims.util.IMSOperType;
            public function onClick():void
            {
                this.map.setCurOper(IMSOperType.ZoomIn);
            }
            public function onClick1():void
            {

```

```

        this.map.setCurOper (IMSOperType.None);
    }
}]>
</mx:Script>
</mx:Application>

```

## 第六章 高级篇

### 第1问 怎样实现自定义控件开发

使用 Flex Builder 可以开发自定义的控件，在新建的 Application 就可以在 Components 选项卡中拖动自己开发的控件。创建自定义控件的方法是在 Flex Builder 的一个工程中选择新建菜单中的 MXML Component 项，然后输入自定义控件名称，选择继承的控件，点击确定即可。

下面是自定义开发的一个实时获取鼠标在地图上位置信息的控件（PositionInfo），源码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<!-- 鼠标的位置信息 -->
<IMSWindow xmlns="zdims.control.*"
implements="zdims.interfaces.control.ICoordsInfoViewer"
    xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute"
    showCloseButton="true"
    x="600"
    y="20"
    height="125"
    width="238"
    title="Mouse Position Infomation"
    borderThicknessRight="0"
    borderThicknessLeft="0"
    styleName="Title"
    creationComplete="init()"
    close="close()">
    <mx:Script>
        <![CDATA[
            import zdims.map.IMSMap;
            import mx.events.FlexEvent;
            import flash.events.MouseEvent;
            import mx.containers.TitleWindow;
            import mx.controls.Alert;
            import mx.controls.Text;

```

```

import mx.events.CloseEvent;

/**
 * 窗口初始化
 */
public override function init():void
{
    super.init();
    if (this.imsmap!=null)
    {

        this.imsmap.addEventListener(MouseEvent.MOUSE_MOVE, this.onMousePos
sChange);
    }
}
/**
 * 移动鼠标修改鼠标位置信息
 */
public function onMousePosChange(e:MouseEvent):void
{
    if (this.imsmap == null)
    {
        Alert.show("map属性为NULL, 请设置PosInfo控件的map属性", "
提示");
        return;
    }
    this.LabelMouseX.text = "Mouse Screen X:" +
this.imsmap.mouseX;
    this.LabelMouseY.text = "Mouse Screen Y:" +
this.imsmap.mouseY;
    var logicCoor:Point =
this.imsmap.screenToLogic(this.imsmap.mouseX, this.imsmap.mouseY);
    this.LableMouseLogicX.text = "Mouse Logic X:" +
logicCoor.x.toFixed(6);
    this.LableMouseLogicY.text = "Mouse Logic Y:" +
logicCoor.y.toFixed(6);
}
]]>
</mx:Script>
<mx:Text id="LabelMouseX"
y="0"
text="Mouse Screen X:0"/>
<mx:Text id="LabelMouseY"
y="20"

```

```

        text="Mouse Screen Y:0"/>
    <mx:Text id="LableMouseLogicX"
        y="40"
        text="Mouse Logic X:0"/>
    <mx:Text id="LableMouseLogicY"
        y="60"
        text="Mouse Logic Y:0"/>
</IMSWindow>

```

## 第2问 怎样实现与 Google Map 的融合

要实现与 Google 地图的叠加显示，需要添加 GoogleMap 图层控件到地图容器 IMSMap 中，设置 MapGis 数据在 GoogleMap 中的开始级数，然后给 IMSMap 的显示范围赋值，范围是 MapGis 数据在 GoogleMap 中的实际范围框。示例代码如下：

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:map="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*"
xmlns:othermap="zdims.othermap.*">
    <map:IMSMap id="imsContainer"
        levelNum="8"
        width="100%"
        height="100%"
        xMinMap="10018754.1713946"
        xMaxMap="15028131.2570919"
        yMinMap="0"
        yMaxMap="5009377.08569732"
        layout="absolute"
        enableAnimate="false">
        <!-- 添加矢量地图文档 -->
        <map:VectorMapDoc mapDocName="hbgoogle.map"
serverAddress="127.0.0.1:5141" alpha="0.5"/>
        <!-- 添加瓦片图层 -->
        <map:TileLayer hdfName="hbgoogle1.HDF"
serverAddress="127.0.0.1:5142" alpha="1"/>

        <!-- 添加GoogleMap图层 -->
        <othermap:GoogleMap mapType="常规地图类型" startLevel="3"
/>
    </map:IMSMap>
</mx:Application>

```

### 第3问 怎样实现负载均衡

要实现服务器负载均衡需要使用请求处理站点程序，该程序在平台安装目录下的 WebSDK\ FlexLib\ relayHandler 文件夹下，有.NET 和 JAVA 两个版本，下面以使用.NET 版本为例演示如何使用实现负载均衡。

第一步：从开始菜单中找到并运行 IMS 集群服务管理器，然后配置瓦片服务和矢量服务器信息。如下图所示：



图6-1 集群服务管理器

第二步：在 RelayHandlerSite 的站点配置文件 Web.config 中修改配置，将 EnableLoadBalance 参数设置为 true 以启用负载均衡，IMSSvrCenterUrl 配置是 IMS 集群服务器的地址，由 IP、端口和虚拟目录名称组成。修改后配置如下：

```
<appSettings>
  <add key="MapGisIMSVectorSvrIP" value="127.0.0.1"/>
  <add key="MapGisIMSVectorSvrPort" value="5141"/>
  <add key="MapGisIMSTileSvrIP" value="127.0.0.1"/>
  <add key="MapGisIMSTileSvrPort" value="5142"/>
  <add key="EnableLoadBalance" value="true"/>
  <add key="IMSSvrCenterUrl"
value="http://localhost:8080/GisCenter"/>
</appSettings>
```

第三步：在新建MXML应用程序中添加一个VectorMapDoc控件和TileLayer控件，把其 serverAddress 属性设置为：

<http://127.0.0.1/RelayHandlerSite/RelayHandler.ashx>即可实现负载均衡。

MXML 应用程序示例代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:map="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*"
xmlns:othermap="zdims.othermap.*">
    <map:IMSMap id="imsContainer"
        levelNum="8"
        width="100%"
        height="100%"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <!-- 添加矢量地图文档 -->
        <map:VectorMapDoc mapDocName="wh.map"
serverAddress="http://127.0.0.1/RelayHandlerSite/RelayHandler.ashx"
alpha="0.5"/>
        <!-- 添加瓦片图层 -->
        <map:TileLayer hdfName="whmap.HDF"
serverAddress="http://127.0.0.1/RelayHandlerSite/RelayHandler.ashx"
alpha="1"/>
    </map:IMSMap>
</mx:Application>
```

## 第4问 怎样使用 Web 转发器转发 Flex 请求到 GIS 服务器

要使用 Web 转发器转发 Flex 请求到 GIS 服务器需要使用请求处理站点程序，该程序在平台安装目录下的 WebSDK\FlexLib\relayHandler 文件夹下，有.NET 和 JAVA 两个版本，下面以使用.NET 版本为例演示如何使用 Web 站点转发 Flex 请求。

第一步：将 RelayHandlerSite 站点发布到 IIS 中。

第二步：在 RelayHandlerSite 的站点配置文件 Web.config 中修改配置。配置参数如下：

```
<appSettings>
    <add key="MapGisIMSVectorSvrIP" value="127.0.0.1"/>
    <add key="MapGisIMSVectorSvrPort" value="5141"/>
    <add key="MapGisIMSTileSvrIP" value="127.0.0.1"/>
    <add key="MapGisIMSTileSvrPort" value="5142"/>
    <add key="EnableLoadBalance" value="false"/>
    <add key="IMSSvrCenterUrl"
value="http://localhost:8080/GisCenter"/>
```



```
</appSettings>
```

提示: MapGisIMSVectorSvrIP是矢量服务主机IP;MapGisIMSVectorSvrPort 矢量服务端口;MapGisIMSTileSvrIP是瓦片服务主机IP;MapGisIMSTileSvrPort瓦片服务端口;EnableLoadBalance是启用负载均衡的标志位,如果需要使用集群服务,该标志位需要设置为true;IMSSvrCenterUrl是集群服务的地址,其地址格式是<http://集群服务主机IP:集群服务端口/集群服务虚拟目录>。

第三步: 在新建MXML应用程序中添加一个VectorMapDoc控件和TileLayer控件, 把它们的serverAddress属性设置

为:<http://127.0.0.1/RelayHandlerSite/RelayHandler.ashx>即可实现请求的转发。

注意: 如果在实际项目开发中需要把通过Web站点转发请求的功能集成到已有的站点中, 以.NET项目为例, 则只需要把RelayHandlerSite文件夹下的RelayHandler.ashx文件拷贝到已有站点中, 引用RelayHandlerSite\bin下面的RelayHandlerLib.dll。在加载站点配置的时候设置类RelayHandlerLib.CRelayHandler中的各项参数(如: RelayHandlerLib.CRelayHandler.MapGisIMSTileSvrIP="127.0.0.1")。然后在MXML应用程序中设置VectorMapDoc、TileLayer或者VectorLayer控件的serverAddress属性为RelayHandler.ashx页面的URL地址即可(如: "<http://127.0.0.1/mywebsite/web/RelayHandler.ashx>" )。RelayHandler.ashx文件名不可以随意更改。

MXML应用程序示例代码如下:

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
creationComplete="init()" layout="absolute" xmlns:map="zdims.map.*"
fontSize="12" xmlns:ns2="zdims.control.*"
xmlns:layer="zdims.control.layer.*"
xmlns:othermap="zdims.othermap.*">
    <map:IMSMap id="imsContainer"
        levelNum="8"
        width="100%"
        height="100%"
        xMinMap="114.125602229914"
        xMaxMap="114.500788705197"
        yMinMap="30.4539323507469"
        yMaxMap="30.8291188260302">
        <!-- 添加矢量地图文档 -->
        <map:VectorMapDoc mapDocName="wh.map"
serverAddress="http://127.0.0.1/RelayHandlerSite/RelayHandler.ashx"
alpha="0.5"/>
        <!-- 添加瓦片图层 -->
        <map:TileLayer hdfName="whmap.HDF"
serverAddress="http://127.0.0.1/RelayHandlerSite/RelayHandler.ashx"
alpha="1"/>
```

```
</map:IMSMap>  
</mx:Application>
```