Svg版本的流程图说明：

在svgworkflow文件夹下面

newFlow.html此页面内容就是编辑流程图，其中编辑流程图有两种方式：一是直接编辑，就是没有xml的情况下，二是有xml，即修改流程图，导入xml进行修改：

方法：initDraw 存在唯一参数 xml ，比如initDraw（x3）,x3是对应的xml，在getxml.js文件中有定义，在系统中，会将xml生成放在隐藏域中，此时的x3 是通过获取隐藏域中的内容去获取xml内容。

这里如果一开始编辑的时候隐藏域中是没有xml的，取值的时候为空，为直接编辑，有值的话就属于流程图修改。

关于属性节点编辑：

过程属性节点(矩形 菱形 等等)编辑 方法在newFlow.html 中的bindToolBar()中

线条节点编辑 在newFlow.html 中的bindLine ()中，

其中上述两个参数绑定方法里面就绑定了几个demo方法，更多的需要后续人员按照要求去绑定。就比一一绑定，这里涉及一些数据是数据库中引用的

里面其他的方法都有说明与使用

readXml.html 是流程图查看，涉及到编辑 其中loadXml为调用方法 第一个参数 为xml的数据，第二个参数为将svg生成在某个元素下面，写入的参数是一个 id的参数 比如 <div id=”canvas”></div> 调用如下 loadXml（x3,”canvas”）

其中 getxml.js 这个文件就是一个演示xml的作用，在系统中是不需要的，直接从隐藏域中引用。

里面还有一个最重要的，编辑好的流程图是要生成xml格式保存到数据库中的，所以在newFlow.html文件中 有个方法很重要 getXml（）方法，最后 return xmlString 这么个参数，其实就是个xml格式的数据，在系统中使用的话，可以修改 return xmlString 为保存在隐藏域中，以便存入数据库中，或者直接异步保存入数据库中，不需要中转一下，这个看具体要求。

其次这里面所有的文件都是静态的html文件，到时候在系统中需要转换成jsp文件，添加一些动态的数据比如xml等，和一些修改。

整个文件我已经移植到 epower 项目中 在workflow 目录下的 svgworkflow目录中，其中我在出去 IE9以上等高级浏览器中做了识别判断，使用svg流程图，仅仅是demo，一些数据还是需要有人去绑定的。

在epower项目 **flowdefineList.jsp** 文件中 做一个编辑流程如的demo引用

其中

<ahref='sampleFlowDefine!getworkflowxml.do?flowCode=${wfFlow.flowCode}&version=0'>编辑流程图草稿</a>

添加onclick方法 editFlow(). 内部是一个判断浏览器的方法 IE低版本的会通过此方法进行跳转，直接跳转，方法editFlow()在**flowdefineList.jsp**的最下面。其他页面也依照此类方式进行修改。