

# 利用 INFOPATH 2007 + VS2005

## 开发 MOSS workflow

( 开发篇 )

撰写者 : moss.net

QQ : 905865083

日期 : 2008-7-17

最近都在学习工作流的开发，和很多朋友一样，刚接触工作流时一头雾水，加上 MOSS2007 复杂的部署，实在是很无助。经过几天的学习，对开发 MOSS 工作流有了初步的了解，在这里结合 MSDN 例子对学习进行一个总结，也希望对初学工作流朋友有个帮助。

### 【事前准备】

开发 MOSS 工作流之前，我们需要下面的开发环境：

- Visual Studio 2005
- Visual Studio 2005 Extensions for Microsoft .NET Framework 3.0

下载地址：

<http://www.microsoft.com/downloads/details.aspx?FamilyID=5d61409e-1fa3-48cf-8023-e8f38e709ba6&DisplayLang=zh-cn>

- Microsoft Office SharePoint Server 2007
- Microsoft Office InfoPath 2007
- Microsoft Office SharePoint Server 2007 SDK (最新 1.3 版本)

下载地址：

<http://www.microsoft.com/downloads/details.aspx?FamilyId=6D94E307-67D9-41AC-B2D6-0074D6286FA9&displaylang=en>

### 【开发步骤】

1. 使用 VS2005 创建 Sharepoint 顺序化工作流类库工程
2. 创建初始化工作流表单 – 用于启动工作流设置
3. 创建任务编辑表单 – 用于用户收集数据

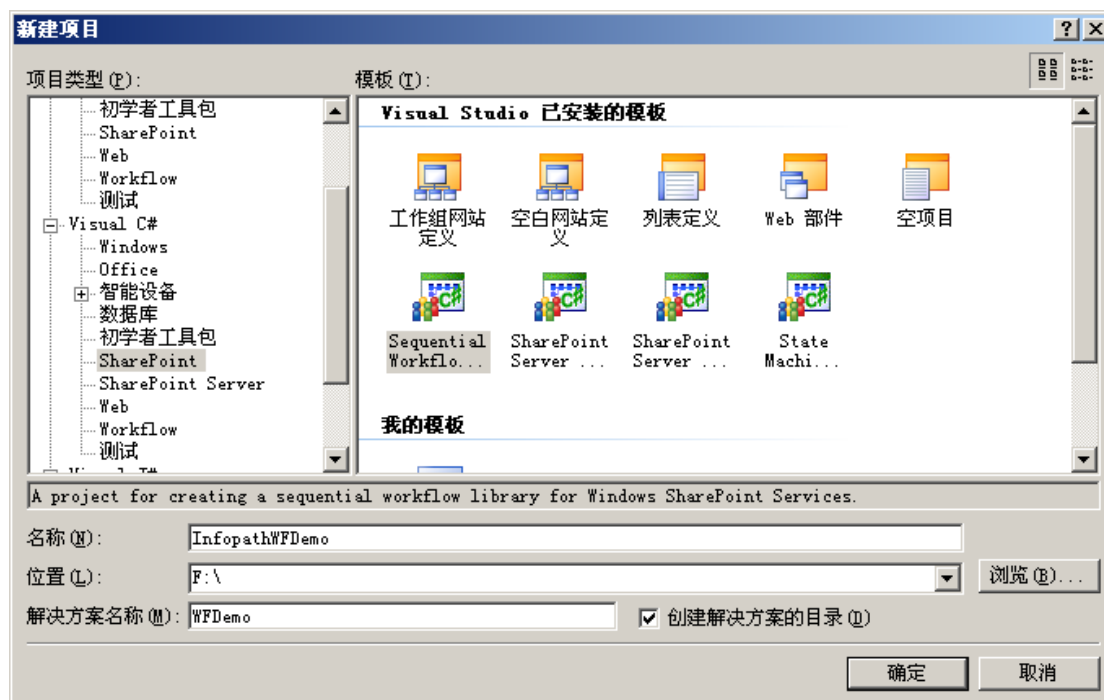
4. 利用 VS2005 workflow 设计器设计 workflow
5. 设置 workflow 中活动相关设置
6. 给活动添加代码
7. 对 workflow 类库进行强签名
8. 部署 workflow

### Step1 【使用 VS2005 创建 Sharepoint 顺序化 workflow 类库工程】

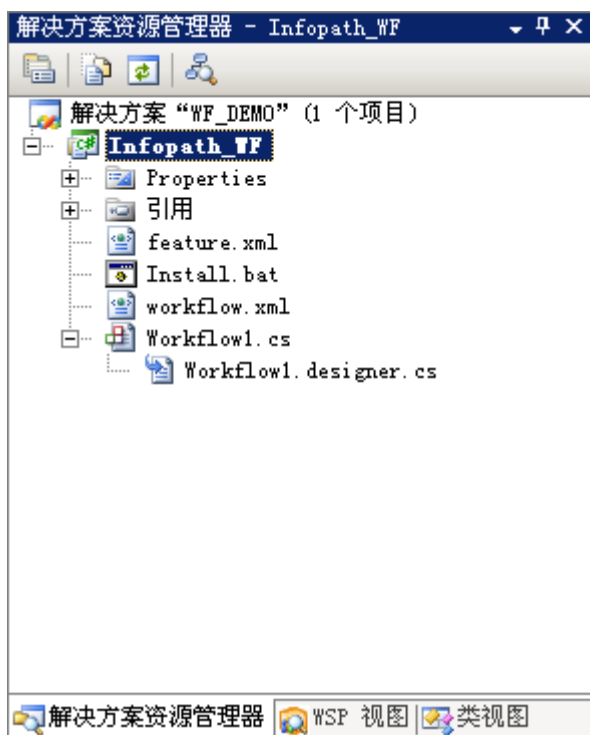
在安装好 MOSS SDK 和 Visual Studio 2005 Extensions for Microsoft .NET Framework 3.0 后，在 VS2005 新建项目模板中就会包含 workflow 模板，这里例子中，我们使用顺序 workflow 模板。

1. 打开 VS2005，新建项目，选择【Sharepoint】

【Sequential Workflow Library】，输入类库名称和路径，点击确定。



2. 通过模板创建工作流项目会自动生成相关文件并添加需要的引用。



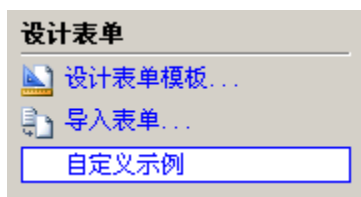
其中，workflow.cs 是工作流的主文件，另外三个文件 feature.xml，install.bat 和 feature.xml 是部署文件，我们在后面会具体介绍。

## Step2 【创建初始化工作流表单】

在工作流中，我们需要利用表单收集用户信息和用户交互，我们可以通过.net form 创建表单。

这里我们使用微软 infopath2007 来创建表单，通过 infopath2007 来创建表单十分方便，简单拖拉操作就可以创建美观实用的表单。在实际开发中，信息工作者可以不需要掌握太多编程技巧就创建好表单，由编程人员创建工作流逻辑代码，由管理员进行部署，很好实现分工。但是，使用 infopath 表单需要有 infopath Service，在 Windows Sharepoint Service 中就不能使用 infopath 表单而只能使用 Form 表单来开发工作流表单。之后，我也会写一些 Form 表单开发工作流的学习笔记。

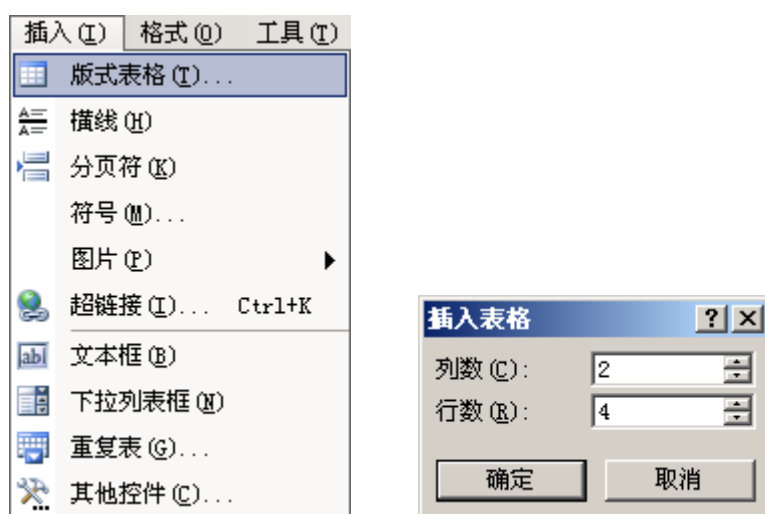
1. 打开 Infopath 2007，点击【设计表单模板】



2. 创建【空白】表单



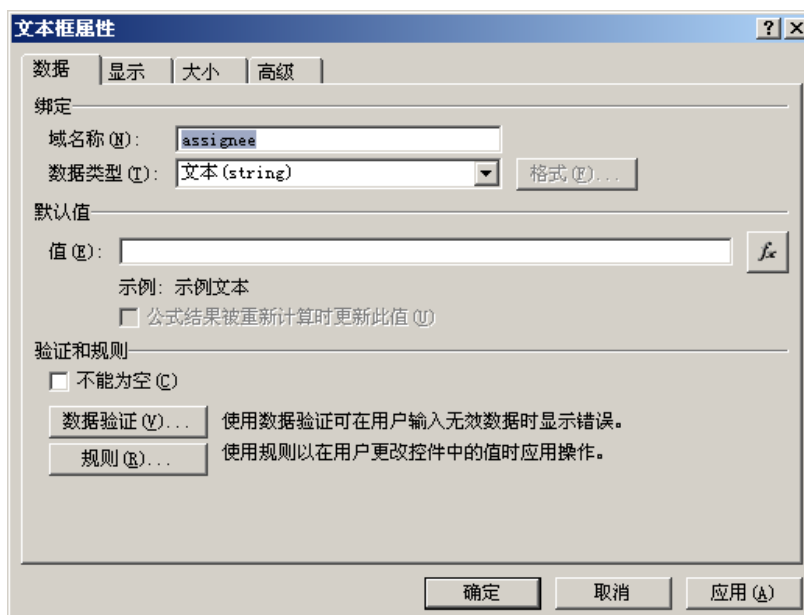
3. 在表单中插入【版式表格】，插入 2 行 4 列表格



4. 通过右边工具栏【控件】选项添加文本框控件和按钮，作成下面的页面

分配给:	
指示:	
备注:	
<input type="button" value="确定"/>	

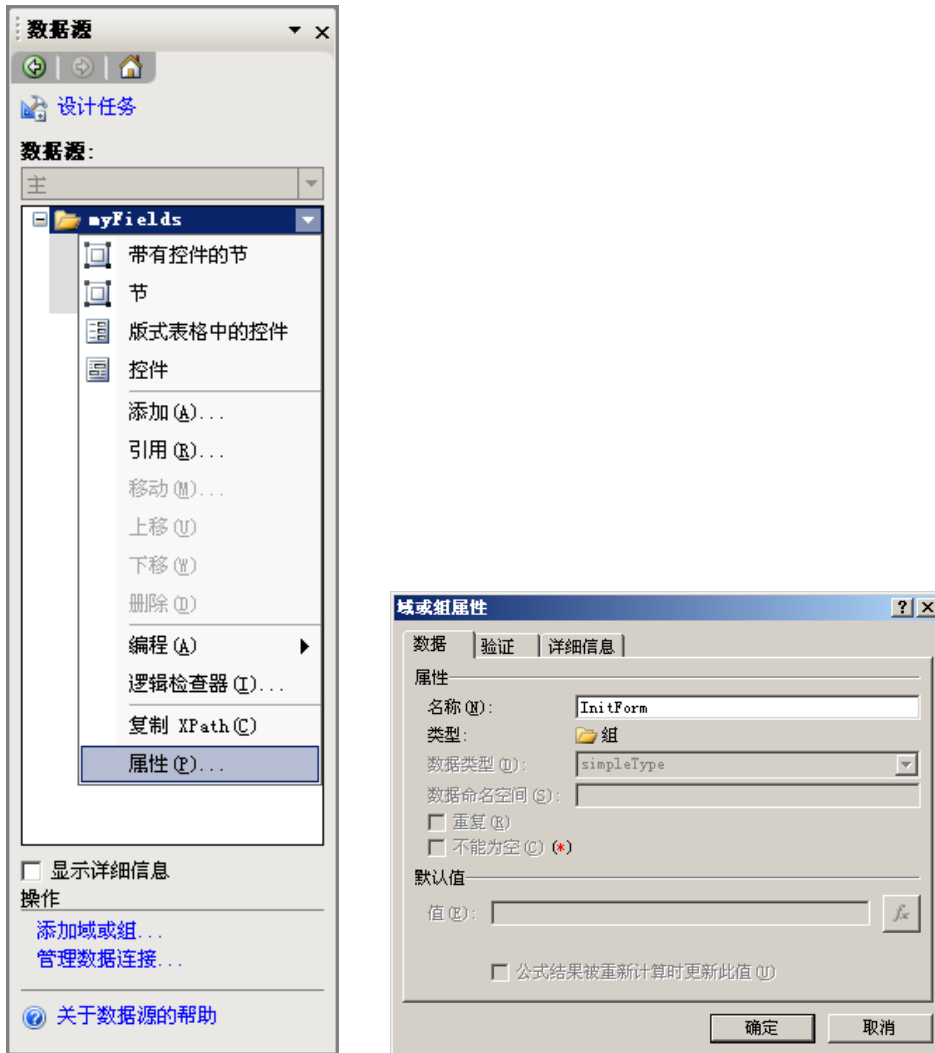
### 5. 双击添加控件修改名称



### 修改三个控件名称

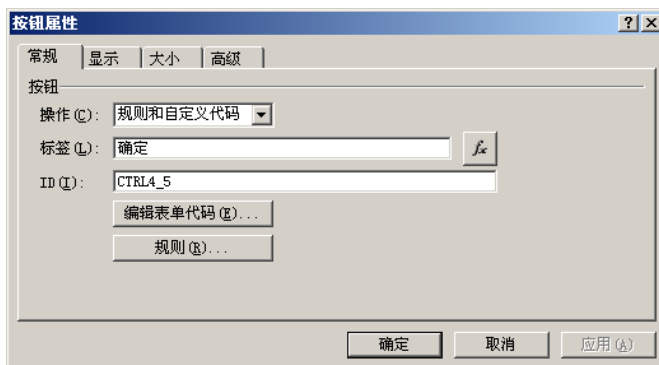
分配给:	<input type="text" value="assignee"/>
指示:	<input type="text" value="instructions"/>
备注:	<input type="text" value="comments"/>
<input type="button" value="确定"/>	

## 6. 在右边工具栏选择【数据源】, 修改主节点【myFields】名称为【InitForm】

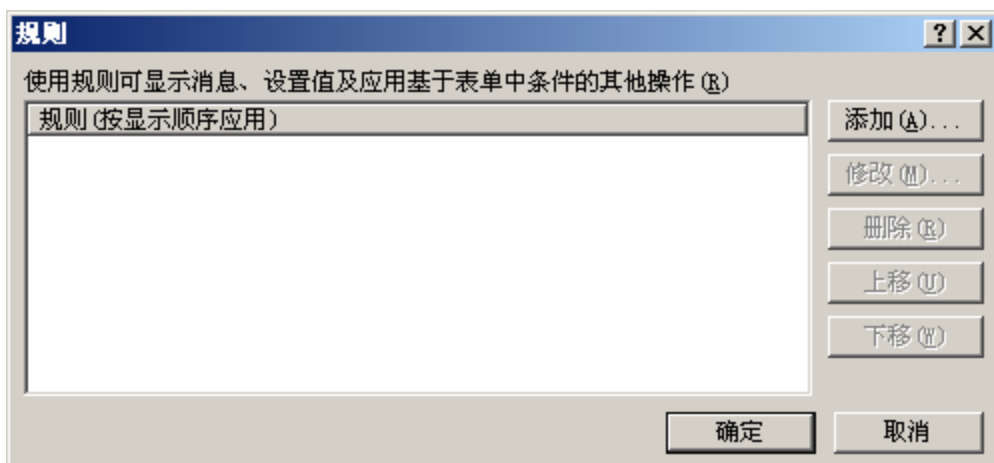


## 7. 添加【确定】按钮点击后操作规则, 也就是点击确定按钮后触发的操作

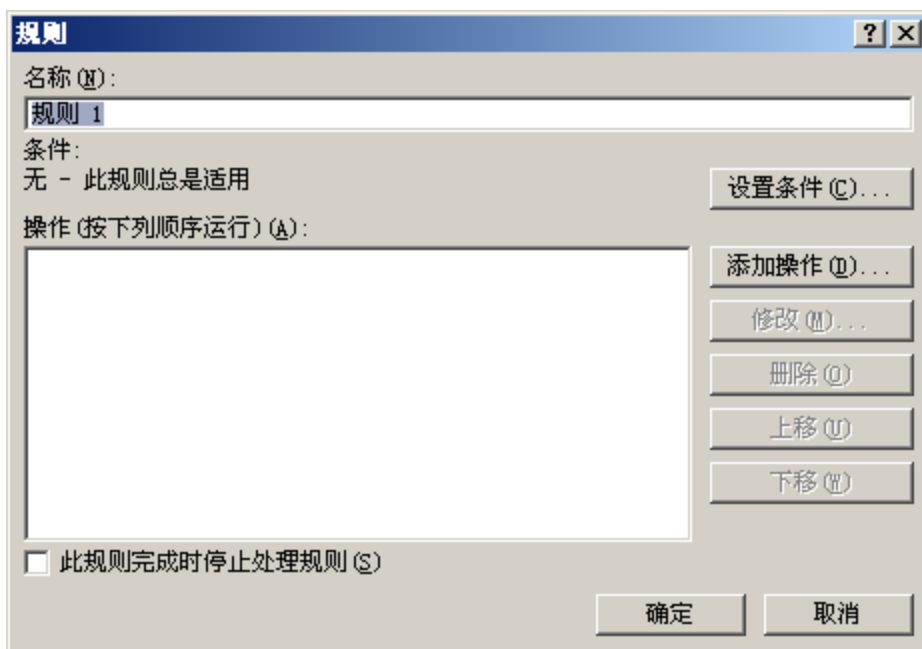
双击【确定】按钮, 弹出下面对话框, 选择【规则】



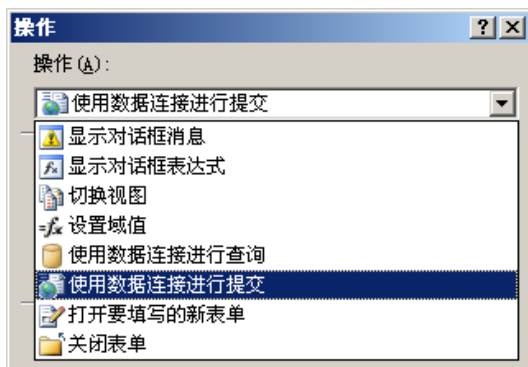
## 添加规则



### 在弹出对话框选择【添加操作】

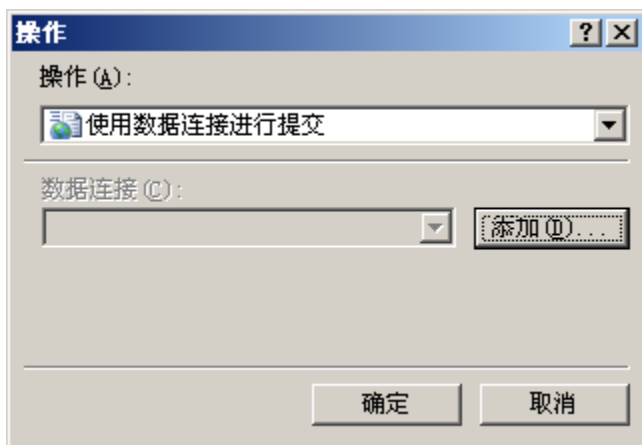


### 在弹出对话框中选择【使用数据连接进行提交】

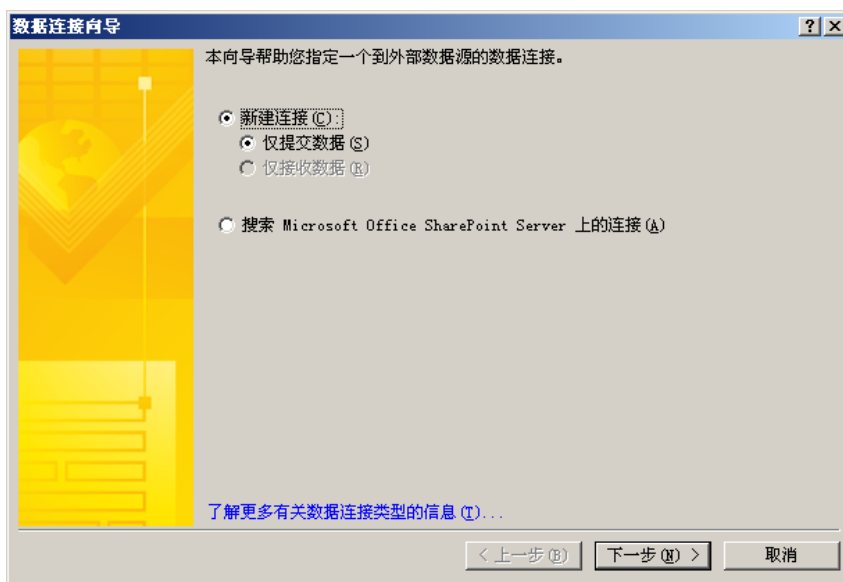




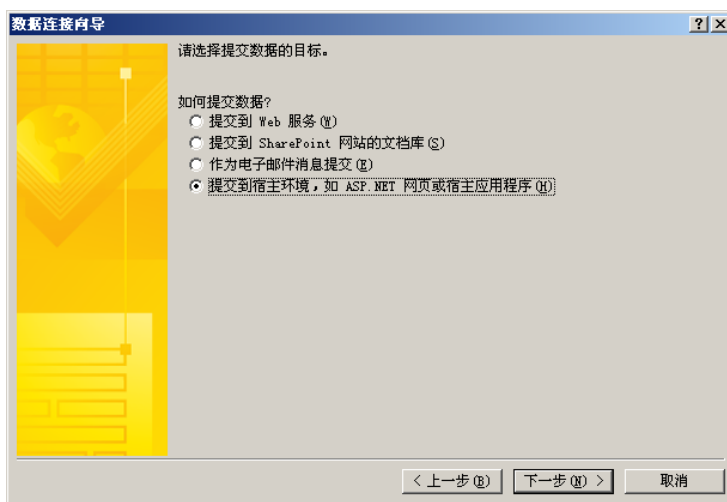
在数据连接中选择【添加】



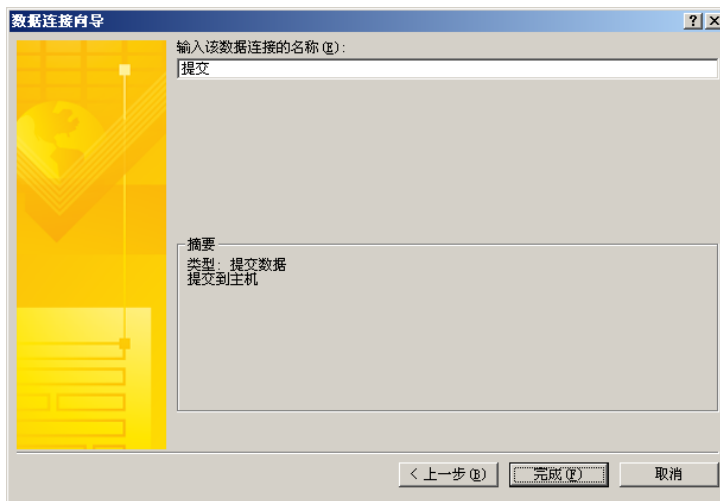
选择【新建连接】【仅提交数据】后，点击【下一步】



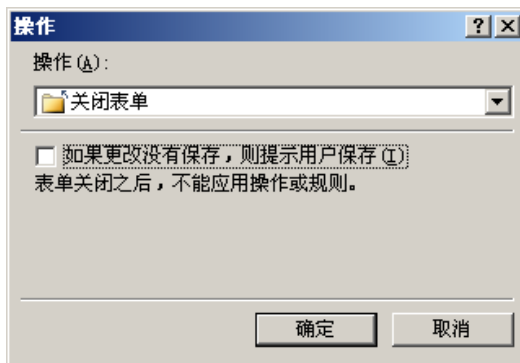
选择【提交到宿主环境,如 ASP.NET 网页或宿主应用程序】后,点击【下一步】



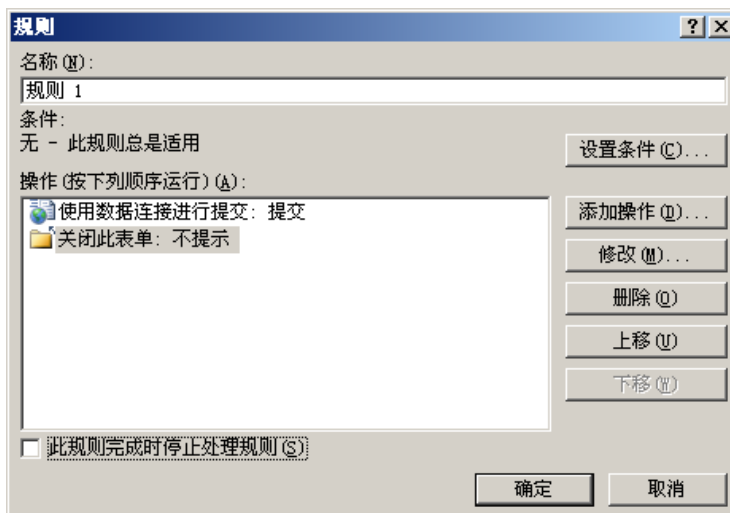
输入连接的名称，点击【完成】



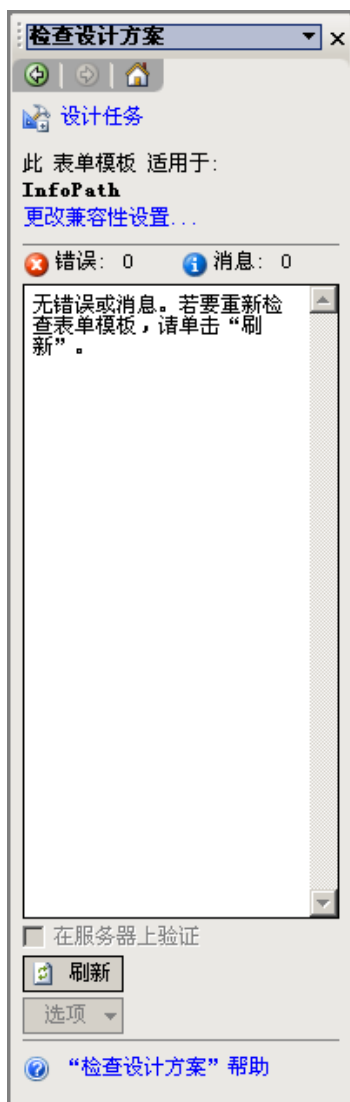
退回到前画面，点击【添加操作】，添加提交数据后的操作，在这里，我们提交数据后应该关闭表单，所以添加【关闭表单】操作



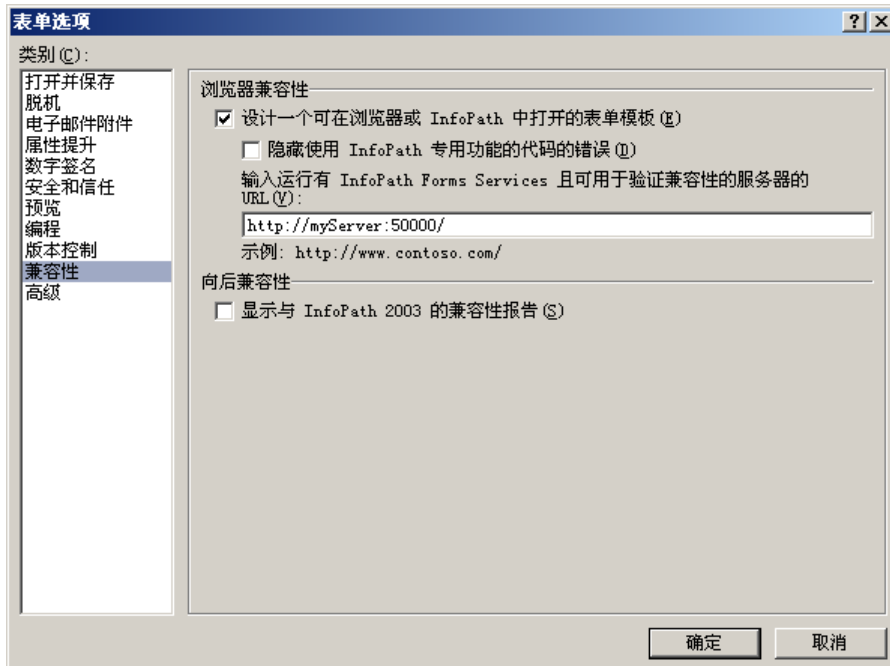
最后在规则对话框中，我们应该看到下面的画面



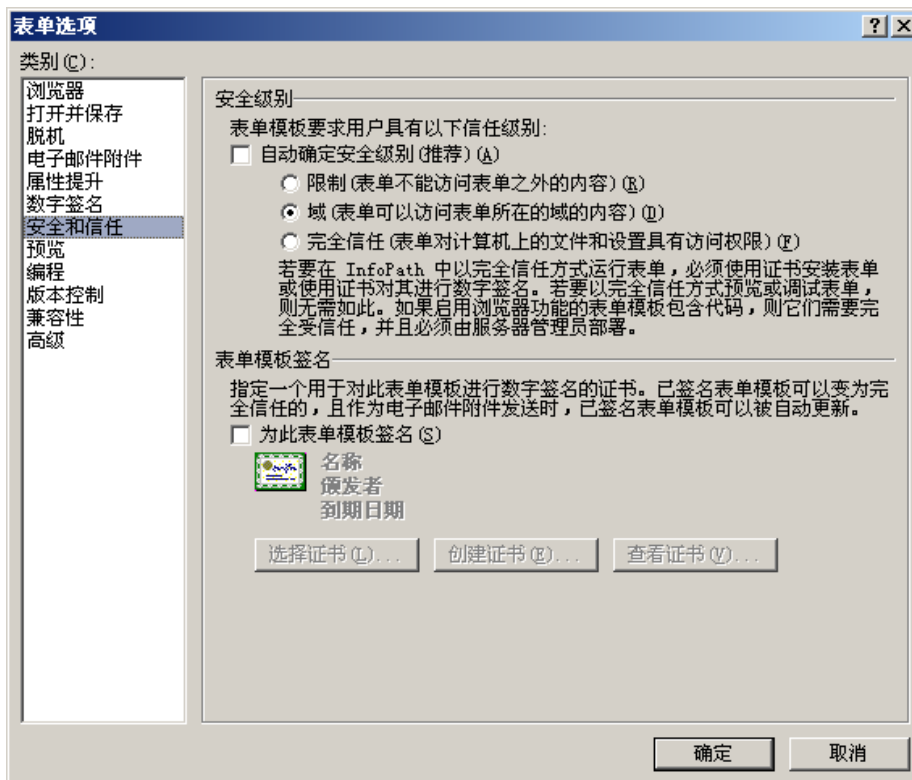
8. 到这里为止，我们就已经创建好工作流初始化表单的画面，接下来我们要对创建好的画面进行兼容性检查。在右边设计任务工具栏中选择【检查设计方案】，点击【更改兼容性设置】



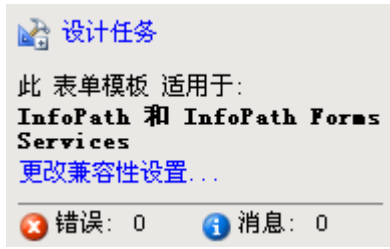
选择【兼容性】选项后，在【设计一个可在浏览器或 Infopath 中打开的表单模板】选项上打勾上，并且输入可以验证表单兼容性的服务器 URL，这里直接输入 MOSS 服务器任意网站 URL 就可以了



选择【安全和信任】选项后，在【安全级别】中选择【域】信任级别

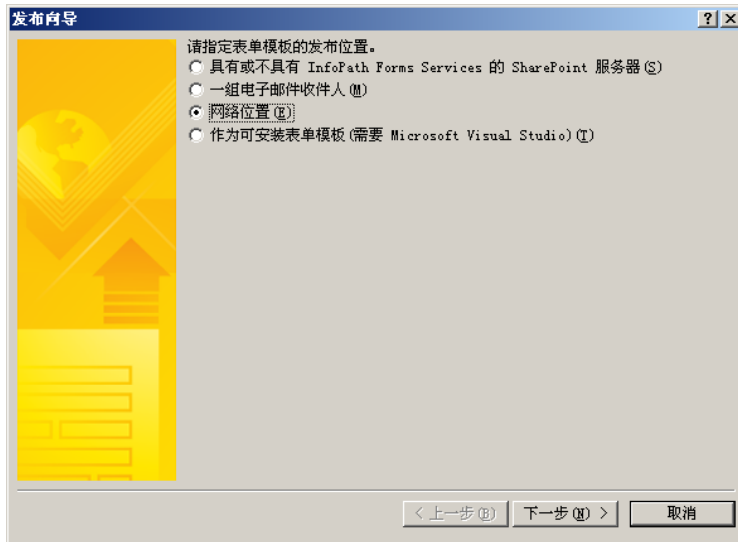


兼容性设置后进行验证，如果没有问题，会出现下面状态栏。

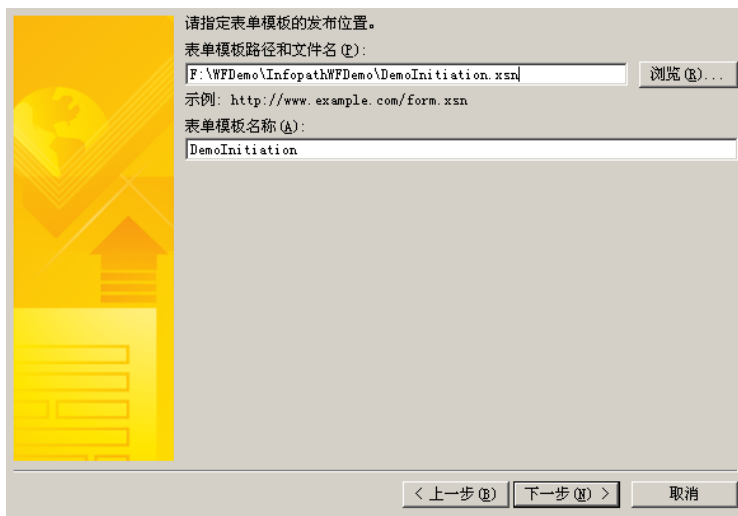


9. 最后，我们需要把创建好的表单发布，我们可以把它发布到 VS2005 工作流工程所在目录

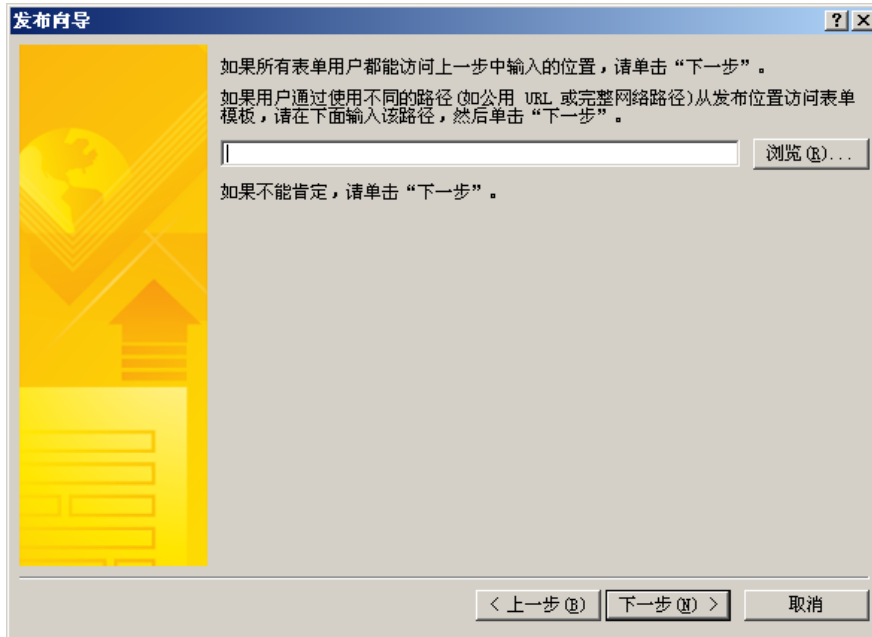
首先保存文件，之后在【文件】菜单选择【发布】，在弹出发布向导菜单选择【网络位置】，点击【下一步】



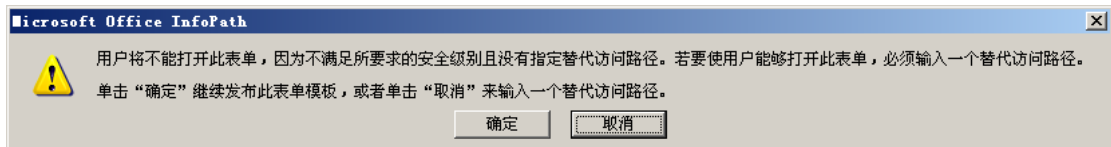
输入发布路径和名称，点击【下一步】



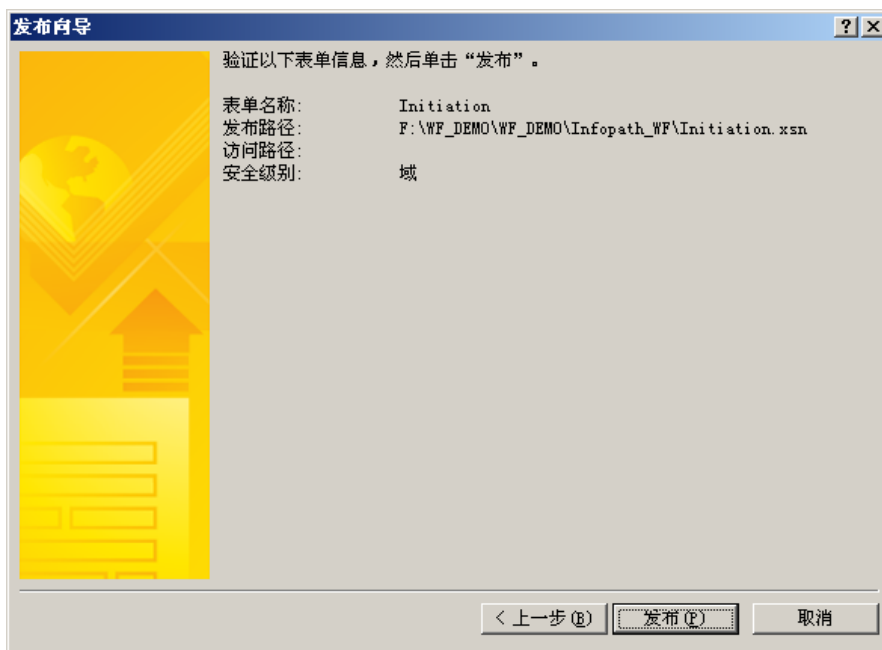
在接下来的对话框中清空备用访问路径，这里如果不清除备用访问路径，在工作流调用时会出现问题。



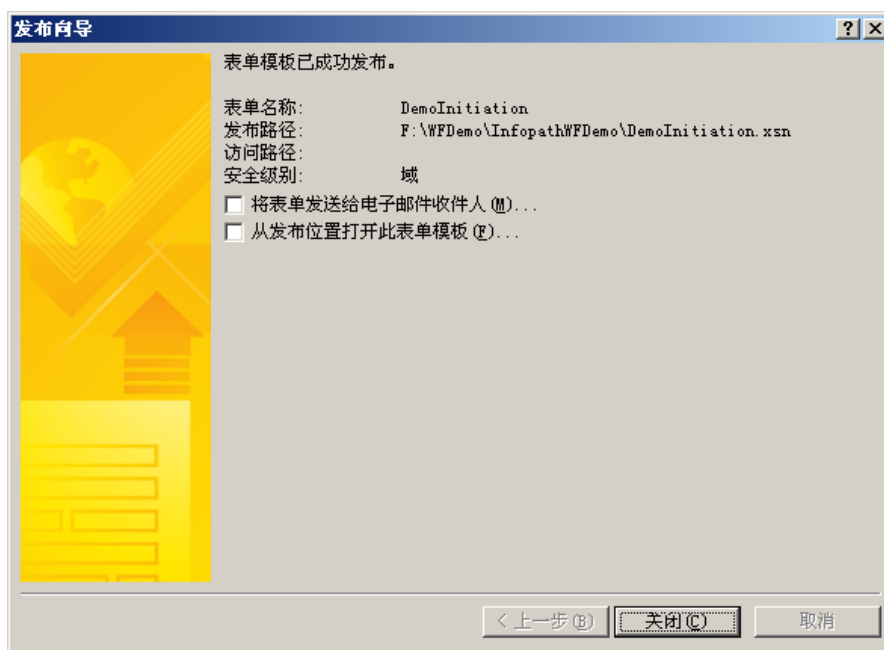
之后会有个警告，直接点击【确定】忽略



最后，点击【发布】，完成表单发布



发布成功后，关闭对话框



10. 由于在工作流中需要取到表单信息，所以我们还需要创建一个帮助类，通过帮助类实例化对象来对表单信息进行操作。这个帮助类可以通过 xsd.exe 来生成

首先，我们需要把创建好的表单保存成源文件。在 INFOPATH 2007 中，选择【文件】，【另存为源文件】



选择保存源文件位置后，点击【确定】，关闭 INFOPATH 2007

打开 VS2005 命令提示方式

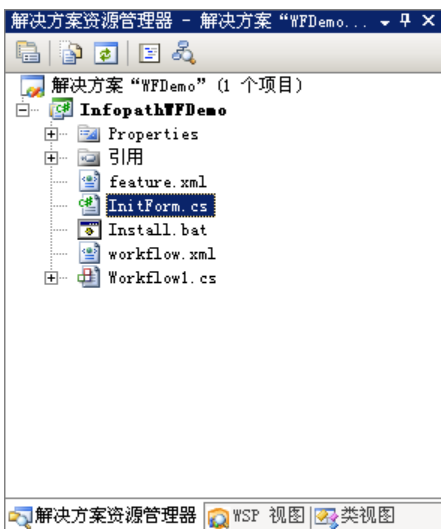


在保存表单源文件位置，输入 `xsd myschema.xsd /c`，保存的表单源文件

默认文件名为 `myschema.xsd`



这样，在源文件路径下就产生一个 CS 文件，`myschema.cs`，我们把名称改成 `InitForm.cs`，并且在刚才创建好的 VS2005 workflow 工程中添加这个文件。





我们可以打开这个类文件，发现表单字段都包含在类中，这样，实例化对象后就可以对表单填写信息进行存取

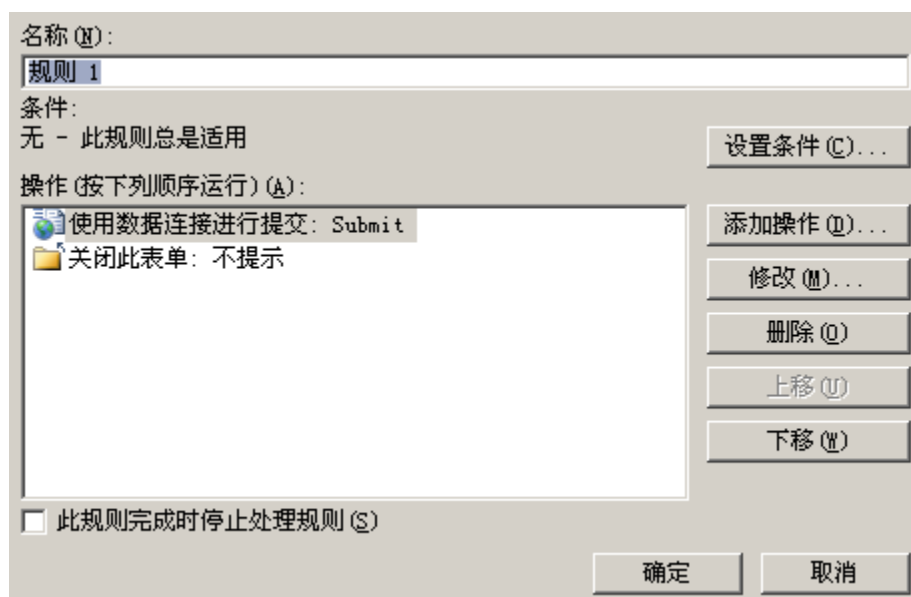
这样，我们对初始化表单工作就完成了。接下来，我们要创建任务编辑表单

### Step3 【创建任务编辑表单】

1. 打开 INFOPATH2007，和之前操作一样，创建空白表单，按照下面例子创建任务编辑表单，由于操作一样，就不详细图例了



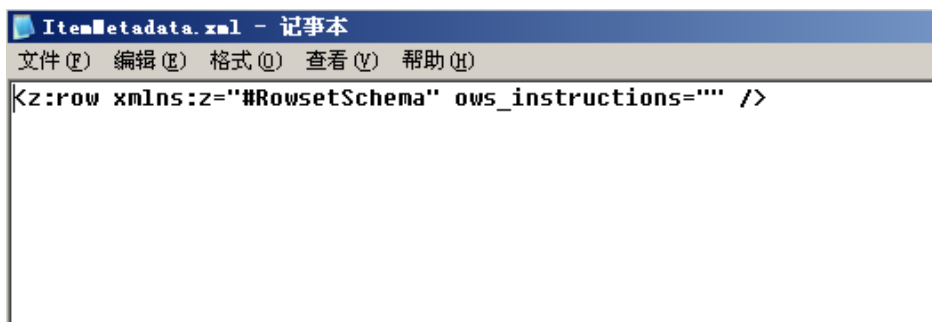
2. 添加【OK】按钮的规则，和前面一样，添加数据连接和关闭表单操作。



3. 由于任务编辑表单需要从初始表单接收数据，我们还需要利用一个 XML 文件，作为它的第二数据源

首先，我们在文本编辑器中创建一个名字为 ItemMetadata.xml 的文件，内

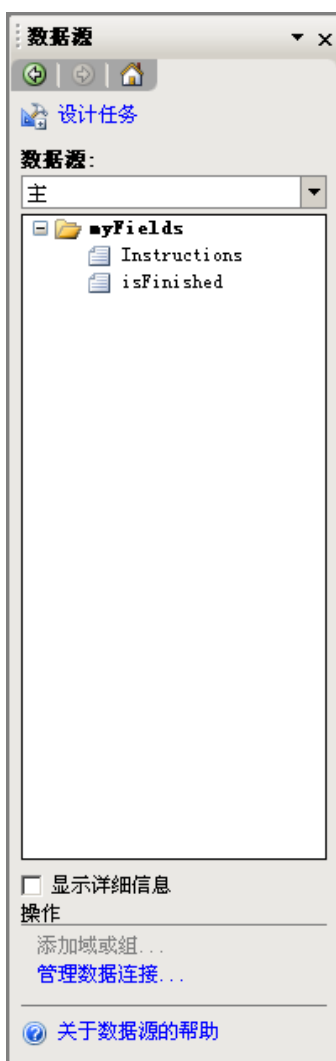
容如下：



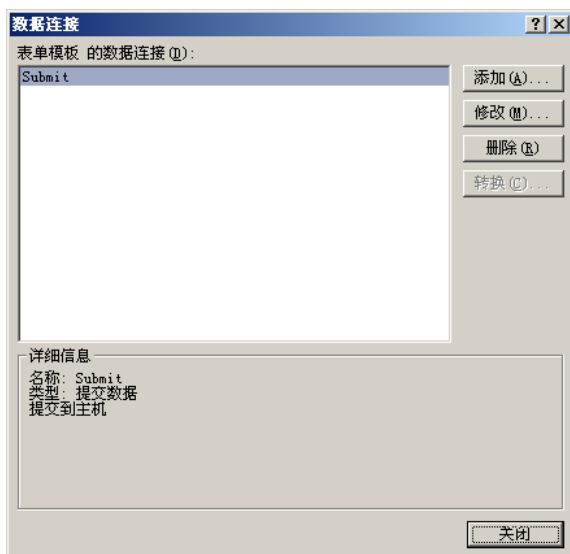
```
ItemMetadata.xml - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<z:row xmlns:z="#RowsetSchema" ows_instructions="" />
```

其中 ows\_ 是前缀，instructions 是我们在表单中需要传递数据的字段名

4. 接下来，我们要绑定数据源到字段上。在右边设计任务工具栏选择【数据源】，在数据源工具栏中，点击【管理数据连接】



5. 在弹出数据连接对话框中点击【添加】



6. 在弹出对话框中，选择【新建连接】【仅接收数据】，点击【下一步】



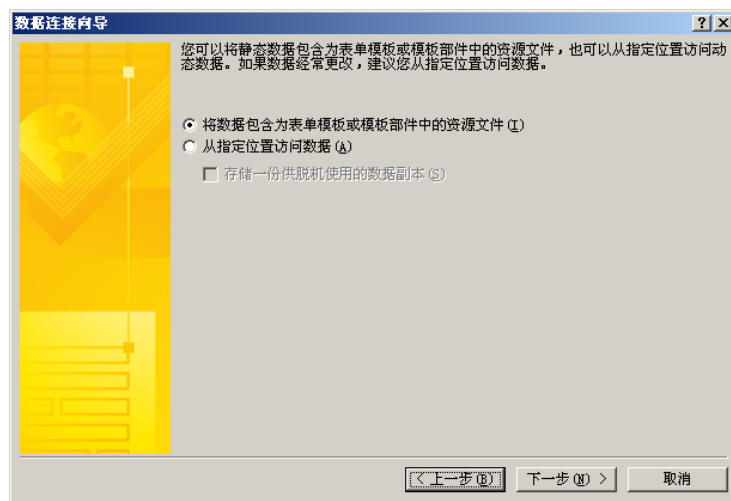
7. 在【数据连接向导】对话框，选择【XML 文档】，点击【下一步】



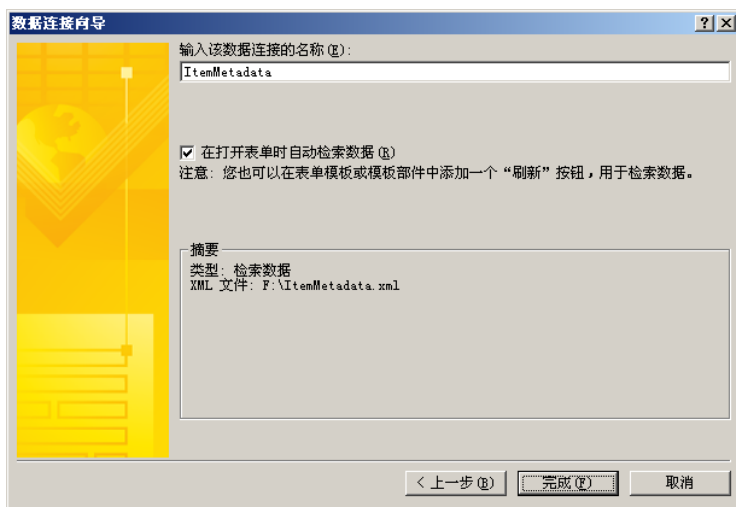
## 8. 选择刚才创建好的 XML 文件，点击【下一步】



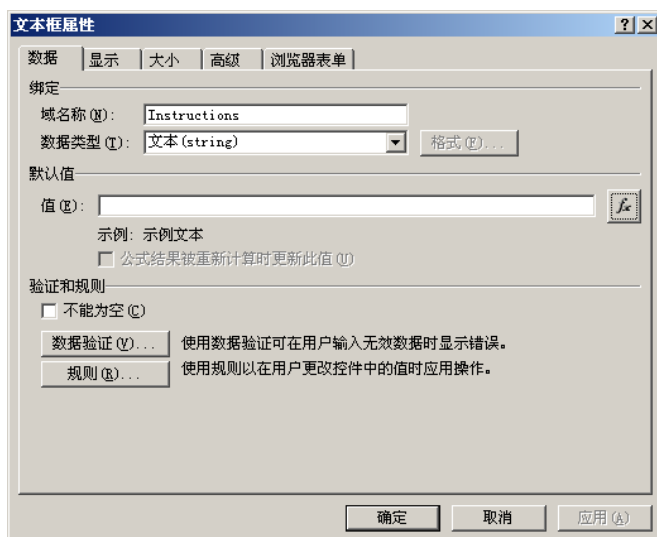
## 9. 选中【将数据包含为表单模板或模板部件中的资源文件】，点击【下一步】



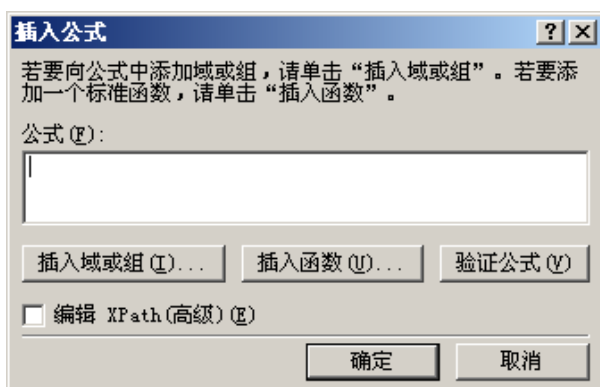
## 10. 输入数据连接名称，选中【在打开表单时自动检索数据】，点击【完成】



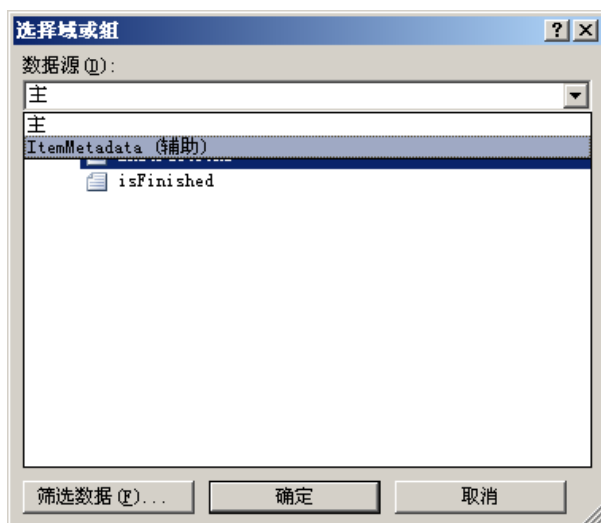
11. 双击表单中需要传递数据的字段控件，这里是 instructions 的文本框控件，  
打开文本框属性对话框，在默认值中点击【fx】按钮



12. 在弹出【插入公式】对话框中点击【插入域或组】



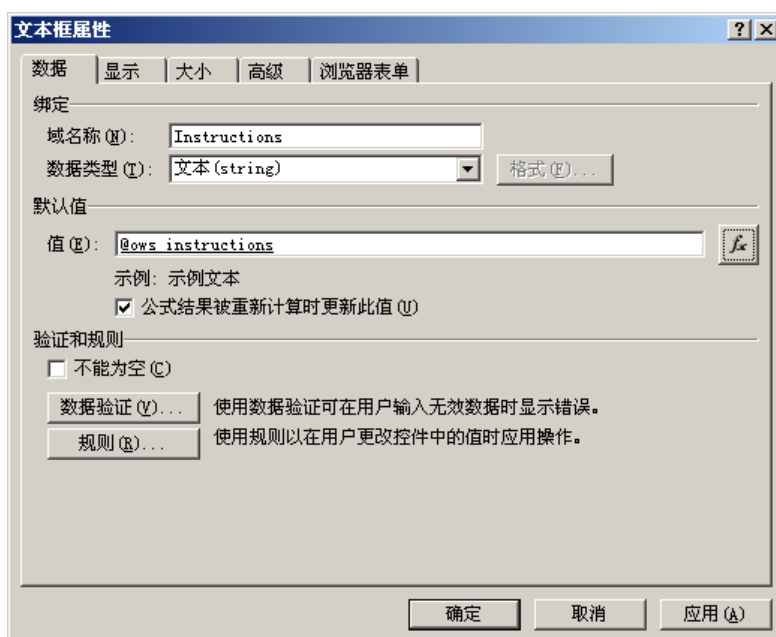
13. 选择刚才创建的 ItemMetadata 数据源



## 14. 选择【:ows\_instructions】 点击【确定】



## 15. 这样，我们就完成数据源绑定



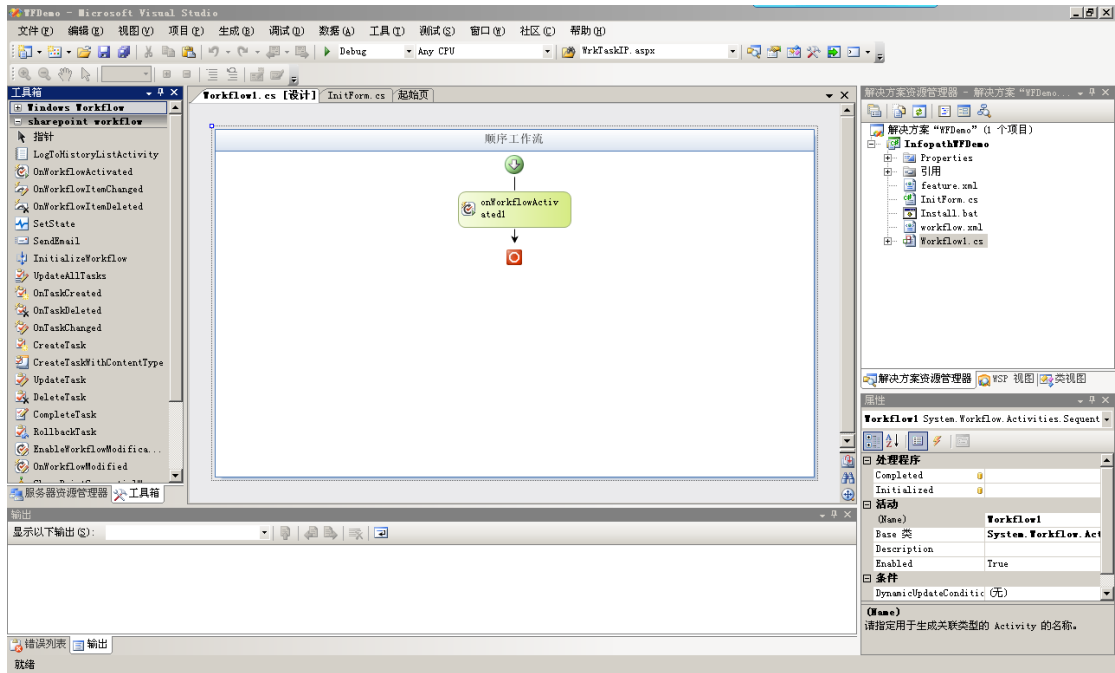
## 16. 对表单兼容性进行检查，步骤和初始化表单兼容性检查相同

## 17. 保存创建好的表单并发布，步骤和初始化表单发布操作相同

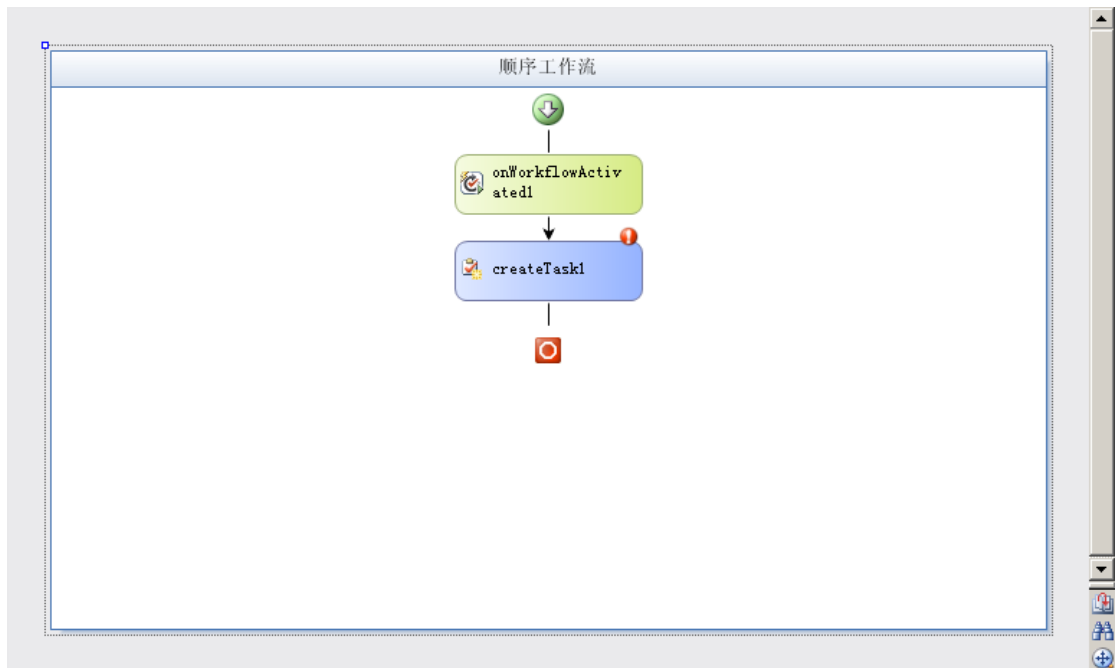
到这里为止，我们对初始化表单和任务编辑表单操作就完成了，接下来，我们要在 VS2005 中创建工作流并把这 2 个表单绑定到工作流上。

## Step4 【利用 VS2005 workflow设计器设计 workflow】

1. 打开 Step1 创建的工作流工程，双击 workflow1.cs 文件，VS2005 会打开 workflow 设计界面，如下

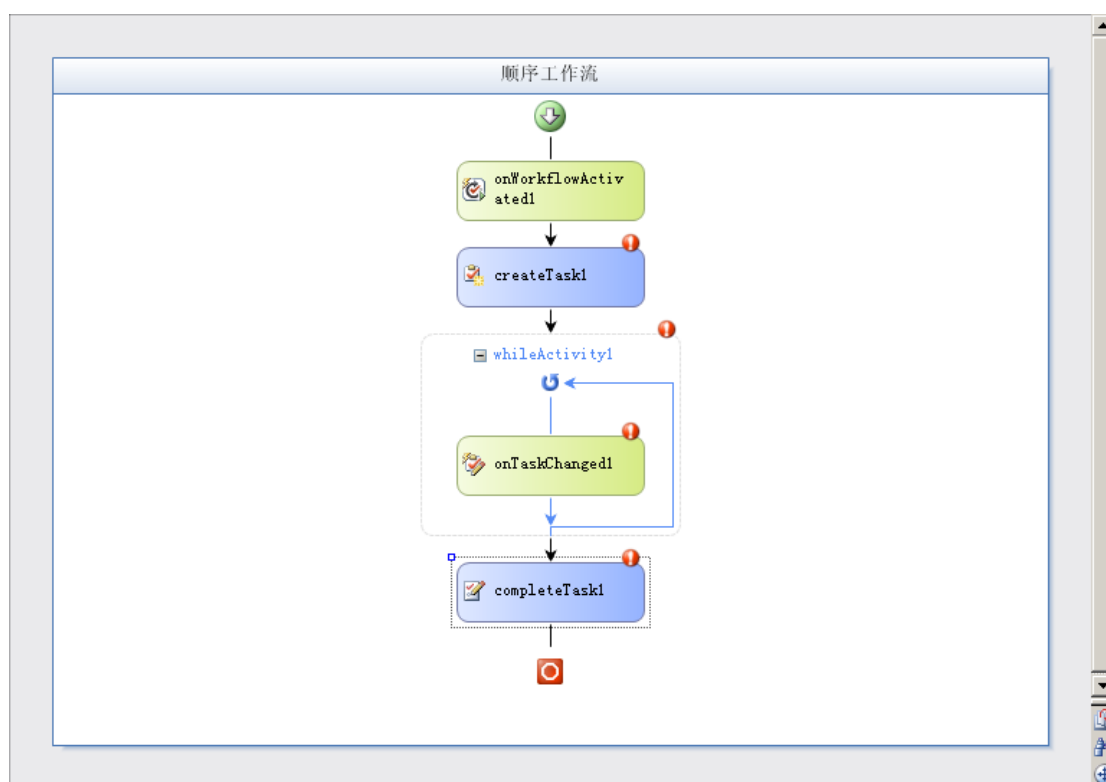


2. 拖动左边工具箱内 workflow 活动部件，添加【CreateTask】活动



3. 操作相同，添加【While】活动部件，并在【While】控件中添加

【OnTaskChanged】，活动部件，最后添加【CompleteTask】活动部件，最后完成 workflow 应该和下图相同



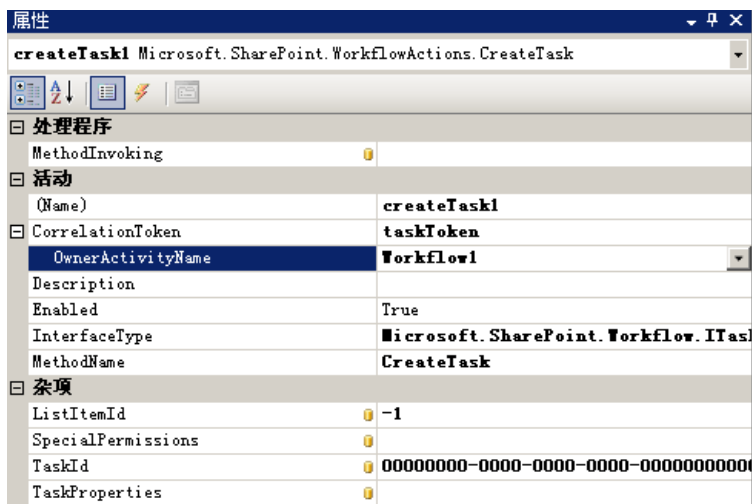
4. 这样，简单 workflow 设计就完成。在这个 workflow 里，首先会创建一个任务给一个用户，并且当用户完成后结束 workflow。在启动 workflow 时，会调用 Step2 创建的初始化表单，在表单上填写分配对象，这样分配对象会收到任务，在点击任务后，会调用任务编辑表单，在 isFinished 上打上勾后代表完成任务，提交后完成任务，最后 workflow 完成

### Step5 【设置 workflow 中活动相关设置】

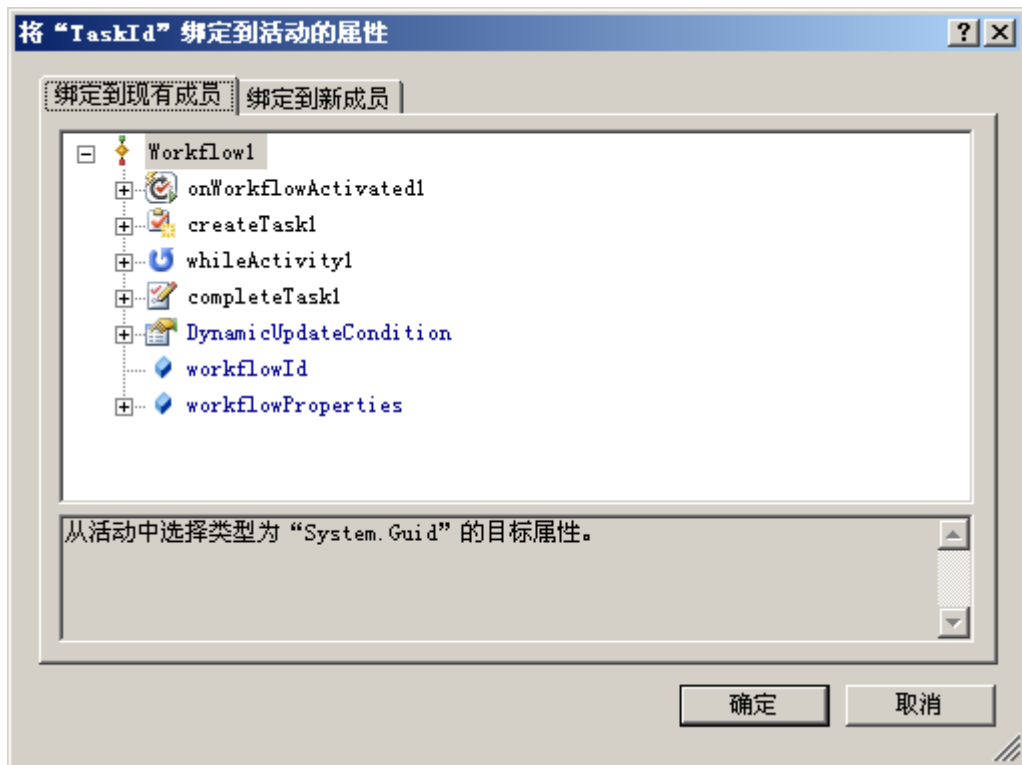
1. 但是这时候，我们看到添加 workflow 活动部件右上角有红色感叹号，这是由于相关属性还没有设置的缘故，为了让这些部件正常工作，接下来我们开始对每个活动部件进行设置



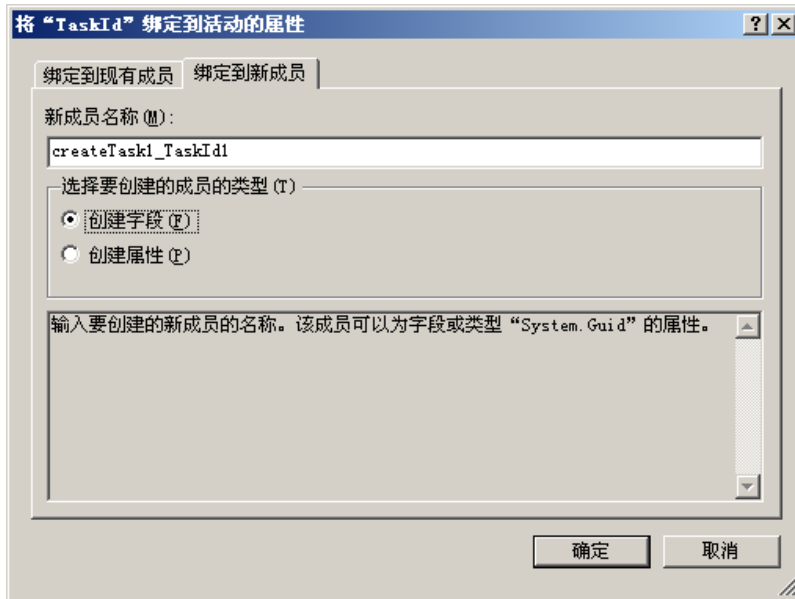
首先,我们选中【CreateTask】活动部件,在属性栏中的【CorrelationToken】中输入【TaskToken】,这里【CreateTask】【OnTaskChanged】和【CompletedTask】三个活动部件代表同一个任务,所以,我们需要设置相同的 TaskToken 给它们,并且它们属于 Workflow1,所以在【OwnerActivityName】中需要选择【Workflow1】



2. 选择【TaskId】后,可以打开下面对话框



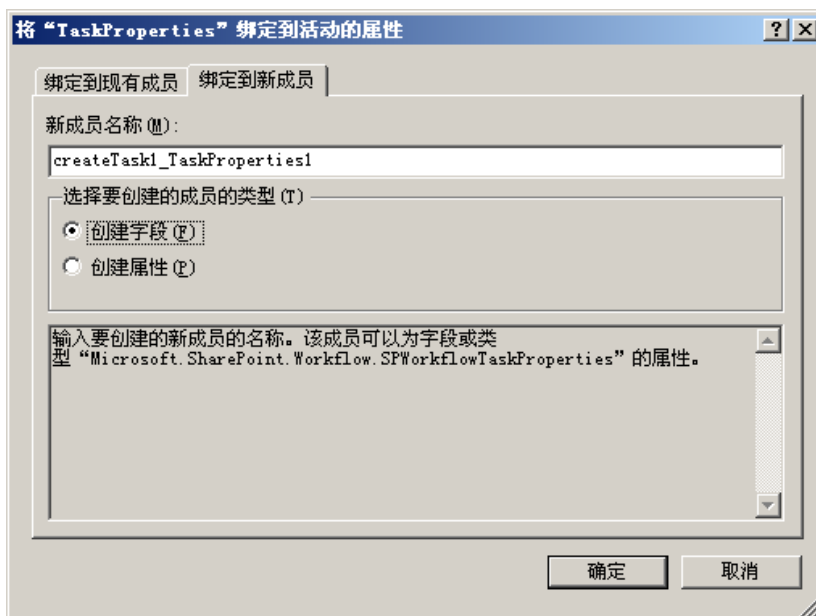
3. 我们需要为任务分配一个成员变量，选择【绑定到新成员】，选择【创建字段】，这里，可以修改变量名称，我们这里选择默认变量名称



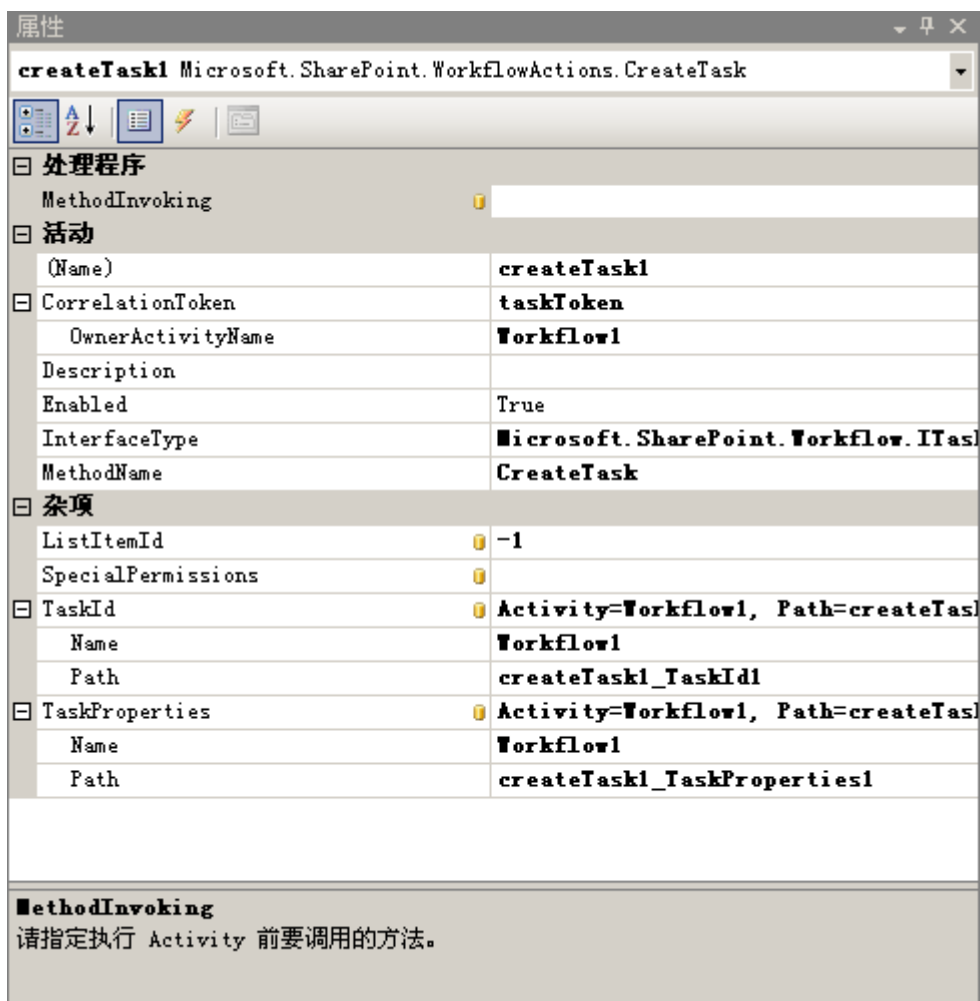
4. 点击确定后，VS2005 会帮你创建好 TaskId 相关信息

<b>TaskId</b>	<b>Activity=Workflow1, Path=createT...</b>
Name	<b>Workflow1</b>
Path	<b>createTask1_TaskId</b>

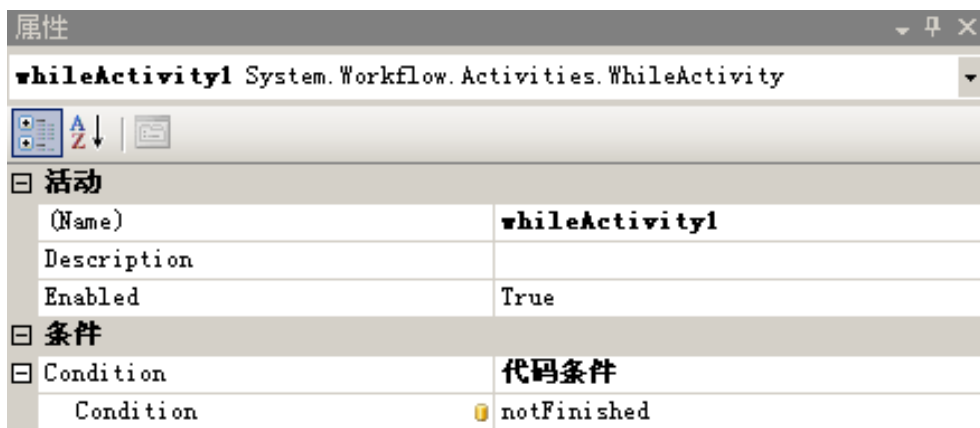
5. 接下来要为任务属性分配一个成员变量，选择【TaskProperties】，弹出对话框中选择【绑定到新成员】，选择【创建字段】



6. 这样，CreateTask 活动部件的设置属性就完成，最后的属性应该如下图：



7. 设置【While】活动部件，选中 While 活动部件，在【Condition】中选择【代码条件】，在展开菜单后，输入方法名【notFinished】，这时候会切换到代码模式，我们先不添加代码，返回到设计模式

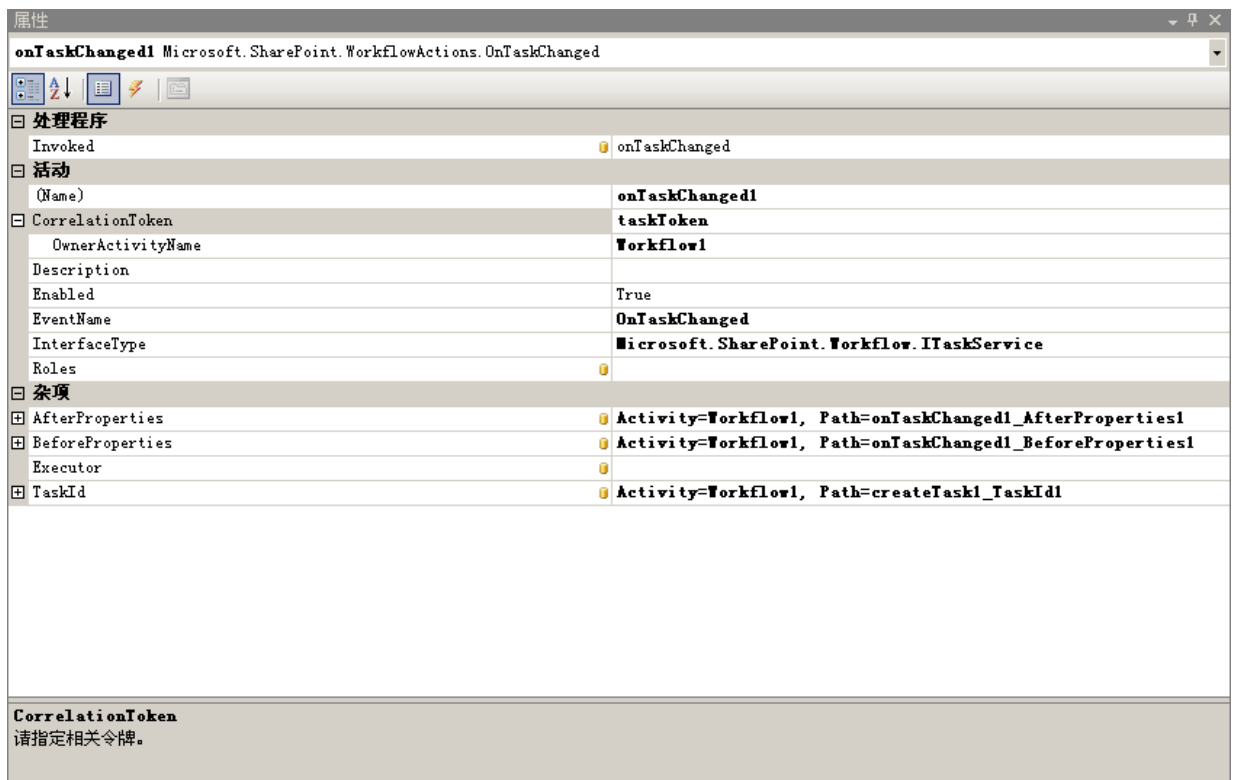


## 8. 在 onTaskChanged 活动部件中

设置 CorrelationToken 属性，选择 CreateTask 中创建的 TaskToken，  
OwnerActivityName 选择 Workflow1

设置 AfterProperties，BeforeProperties 属性，给这两个属性分配新成员  
变量

在 TaskId 上绑定到 CreateTask 活动部件中创建的 TaskId 变量 这里注意，  
因为是相同任务，所以这里不是创建新变量，而是绑定到现有成员，设置后  
画面如下：

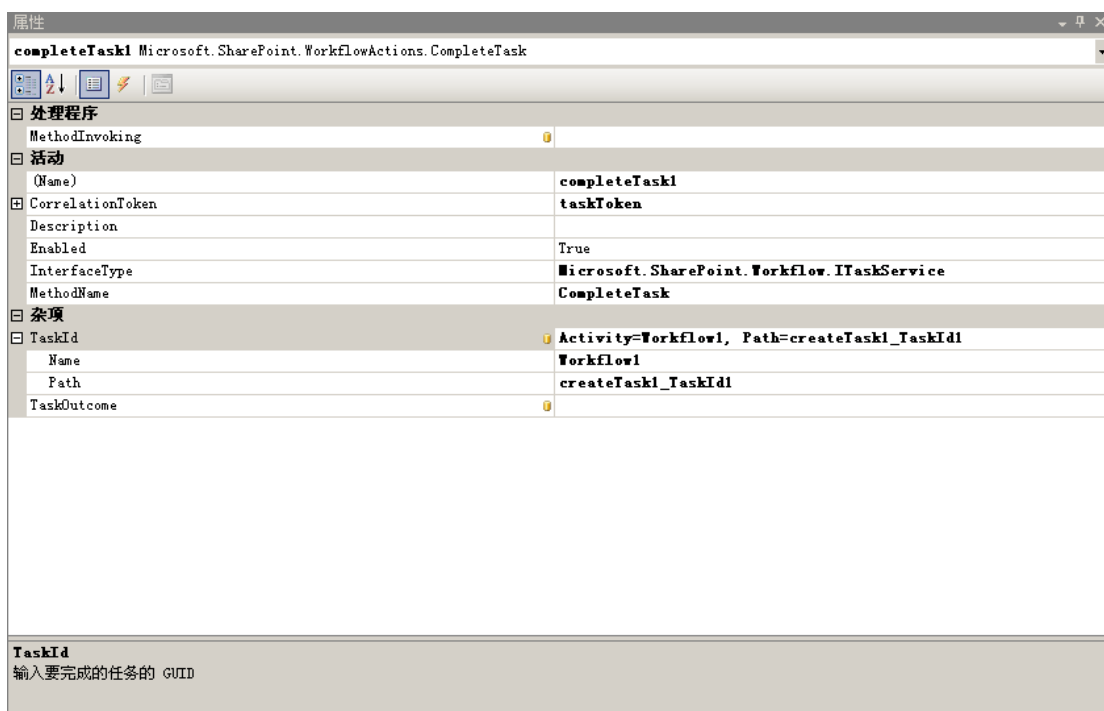


## 9. 最后，设置 CompletedTask 活动部件

设置 CorrelationToken 属性，选择 CreateTask 中创建的 TaskToken，  
OwnerActivityName 选择 Workflow1

在 TaskId 上绑定到 CreateTask 活动部件中创建的 TaskId 变量 这里注意，

因为是相同任务，所以这里不是创建新变量，而是绑定到现有成员，设置后画面如下：



10. 到此， workflow 活动部件设置就完成，接下来我们需要对这些部件添加代码

## Step6 【给活动添加代码】

1. 首先，在设计模式单击右键，选择【查看代码】，添加需要定义成员变量

```
private String assignee = default(String);
private String instructions = default(String);
private String comments = default(String);
private bool isFinished;
```

2. 双击【onWorkflowActivated1】，这时候会进入到代码模式，添加代码：

```
//取得 Workflow ID
workflowId = workflowProperties.WorkflowId;
//表单反序列化，通过InitForm对象取得表单信息
XmlSerializer serializer = new XmlSerializer(typeof(InitForm));
XmlTextReader reader = new XmlTextReader(new
System.IO.StringReader(workflowProperties.InitiationData));
InitForm initform = (InitForm)serializer.Deserialize(reader);
assignee = initform.assignee;
instructions = initform.instructions;
comments = initform.comments;
```

### 3. 回到设计模式，双击【createTask1】，进入到代码模式，添加代码：

```
//创建任务，分配给初始表单中assignee字段的用户并传递表单信息
createTask1_TaskId1 = Guid.NewGuid();
createTask1_TaskProperties1.Title = "Demo Task";
createTask1_TaskProperties1.AssignedTo = assignee;
createTask1_TaskProperties1.Description = instructions;
createTask1_TaskProperties1.ExtendedProperties["comments"] = comments;
createTask1_TaskProperties1.ExtendedProperties["instructions"] =
instructions;
```

### 4. 回到设计模式，双击【onTaskChanged1】，进入到代码模式，添加代码：

```
isFinished =
bool.Parse(onTaskChanged1_AfterProperties1.ExtendedProperties["isFini
shed"].ToString());
```

### 5. 最后，在代码模式中，查看刚才在【While】活动部件中创建的【notFinished】

方法，在方法中添加代码：

```
e.Result = !isFinished;
```

## Step7 【对 workflow 类库进行强签名】

### 1. 完成代码后，在 VS2005 的工具栏菜单中选择【项目】，【属性】



## 2. 为程序集添加签名



## 3. 编译成 DLL 文件后，就可以进行最后的部署

部署 workflow 也是个挺复杂的事情，这几天我会再做一份详细文档放到网络上，开发说明有不足之处，欢迎大家指点。

欢迎大家来 <http://bbs.winos.cn>

论坛一起讨论和学习 MOSS