Scut快速开发（2）

Python脚本开发

# 开发环境

**需要安装的软件**

1. SQL Server2005
2. VS2010开发工具
3. CodeSmith 5.2

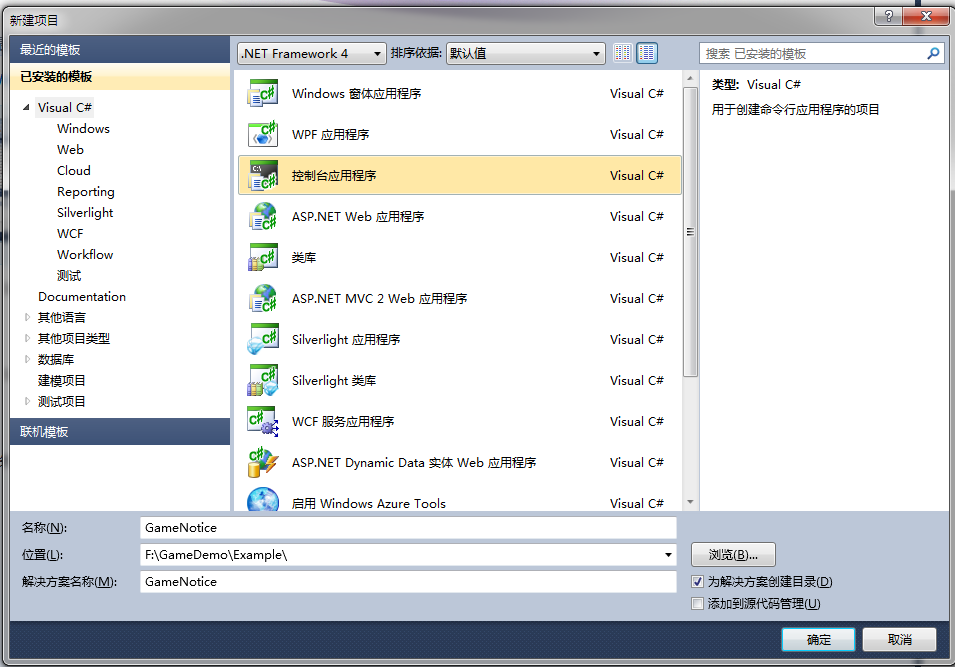
**工具**

1. 协议工具

# 游戏公告

## 创建项目

打开VS2010，新建一个控制台项目命名为GameNotice，并设置项目属性的目标框架为Net Framework 4.0；如图：



**组件引用**

|  |  |
| --- | --- |
| ****项目**** | **引用路径** |
| **HostServer** | **Plugin\Newtonsoft.Json.dll**  **Plugin\NLog.dll**  **Plugin\Redis\protobuf-net.dll**  **Plugin\Redis\** **ServiceStack.Common.dll**  **Plugin** **\Python\IronPython.dll**  **Plugin** **\Python\IronPython.Modules.dll**  **Plugin** **\Python\Microsoft.Dynamic.dll**  **Plugin** **\Python\Microsoft.Scripting.dll**  **Framework\V5.1\ZyGames.Framework.Common.dll**  **Framework\V5.1\ZyGames.Framework.dll Framework\V5.1\ZyGames.Framework.Plugin.dll**  **Framework\V5.1\ZyGames.Framework.RPC.dll**  **Game\V2.6\ZyGames.Framework.Game.Lang.dll**  **Game\V2.6\ZyGames.Framework.Game.dll**  **Game\V2.6\** **ZyGames.Framework.Game.Contract.dll** |

（注：检查**Python的DLL组件引用属性面版， ”复制本地“属性为True**）

增加GameHostApp类，继承至GameHost基类，在Main方法中调用

|  |
| --- |
| // Program.cs  GameHost.Start(new GameHostApp()); |

增加GameHostApp类中的OnRequested的方法处理代码：

|  |
| --- |
| // GameHostApp.cs  protected override void OnRequested(HttpGet httpGet, IGameResponse response)  {  try  {  var actionId = httpGet.GetString("ActionID").ToInt();  var uid = httpGet.GetString("uid");  Console.WriteLine("Action{0} from {1} {2}", actionId, httpGet.RemoteAddress, uid);  ActionFactory.Request(httpGet, response, null);  }  catch (Exception ex)  {  Console.WriteLine("出错：{0}", ex);  }  } |

项目层次结构划分，增加Model目录存储数据实体类和PyScript目录存放脚本文件；**修改App.config配置:**

|  |
| --- |
| **<?xml version="1.0"?>**  **<configuration>**  **<appSettings>**  **<!--服务监听端口-->**  **<add key="Game.Port" value="9001" />**  **<!--Code：游戏代码；ServerId：游戏分区代码-->**  **<add key="Product.Code" value="1"/>**  **<add key="Product.ServerId" value="1"/>**  **<!--Python脚本配置**  **Python\_IsDebug：是否开启调试功能**  **PythonRootPath：脚本路径，相对于程序运行目录**  **-->**  **<add key="Python\_IsDebug" value="true"/>**  **<add key="PythonRootPath" value="..\..\PyScript"/> </appSettings>**  **<connectionStrings>**  **<add name="GameData" providerName="" connectionString="Data Source=.;Initial Catalog=GameData;Integrated Security=True;"/>**  **</connectionStrings>**  **<startup>**  **<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>**  **</startup>**  **</configuration>** |

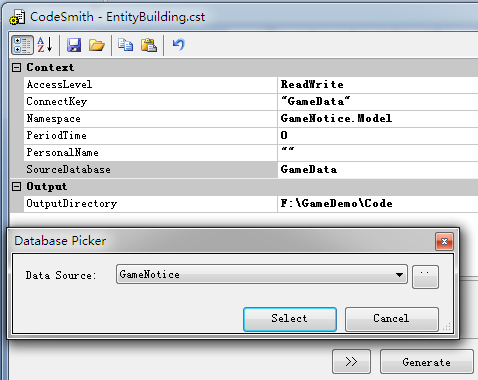
## 创建数据库

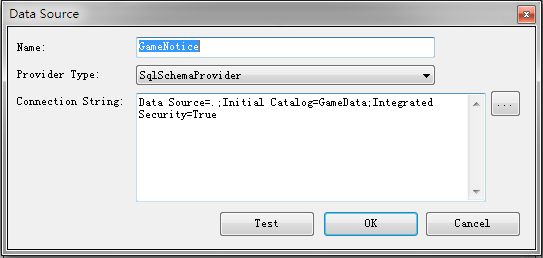
使用SQL Server2005建立一个GameData库，并建立一张公告（Notice）表，字段如下：

|  |
| --- |
| Id int primary key，  Title Varchar(50)，  Content Varchar(200)，  CreateDate DateTime |

**数据实体生成**

打开CodeSmith工具的EntityBuilding模板，生成Notice表的数据实体，如下：





生成成功后，打开输出目录，将Notice文件复制到项目中Model目录下，并在项目中包含此文件；

## 定义协议接口

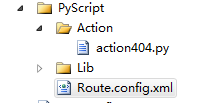
打开协议生成器工具，增加一个“GameNotice”项目方案，接着在增加公告信息（接口编号2001）协议，请求参数和下发参数；如图：



## 编写脚本

**导入脚本库**

复制Scut/PythonLib/目录下所有文件到项目PyScript目录下，并包括到项目中，并修改**路由配置表（**Route.config.xml）的Python安装类库路径；如图：



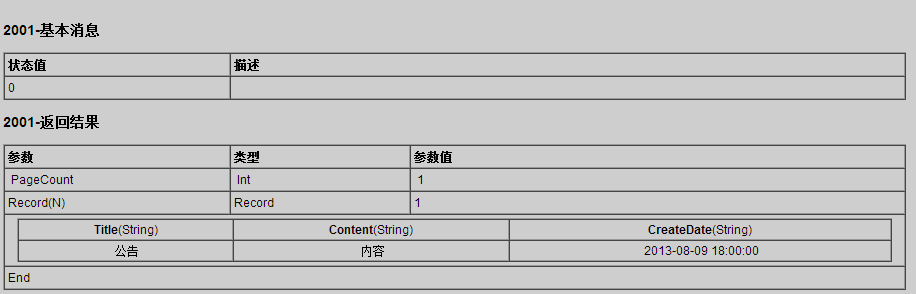
**新增脚本**

**在Action目录下创建脚本协议接口action2001.py，把协议工具生成的2001的脚本代码复制到action2001.py文件；接着将2001接口增加到路由表中，设置”** **ignoreAuthorize”为true；以下是需要修改的部分代码：**

|  |
| --- |
| **#注册DLL和引用命名空间**  **import clr, sys**  **from action import \***  **clr.AddReference('ZyGames.Framework')**  **clr.AddReference('ZyGames.Framework.Common')**  **clr.AddReference('ZyGames.Framework.Game')**  **clr.AddReference('GameNotice')**  **from ZyGames.Framework.Common import \***  **from ZyGames.Framework.Cache.Generic import \***  **from ZyGames.Framework.Game.Cache import \***  **from ZyGames.Framework.Game.Service import \***  **from GameNotice.Model import \***  **... ...**  **def takeAction(urlParam, parent):**  **actionResult = ActionResult()**  **noticeList = ShareCacheStruct[Notice]().FindAll()**  **result = MathUtils.GetPaging[Notice](noticeList, urlParam.PageIndex, urlParam.PageSize)**  **if result:**  **actionResult.dsItemCollect = result[0]**  **actionResult.PageCount = result[1]**  **return actionResult**  **def buildPacket(writer, urlParam, actionResult):**  **writer.PushIntoStack(actionResult.PageCount)**  **writer.PushIntoStack(len(actionResult.dsItemCollect))**  **for info in actionResult.dsItemCollect:**  **dsItem = DataStruct()**  **dsItem.PushIntoStack(info.Title)**  **dsItem.PushIntoStack(info.Content)**  **dsItem.PushIntoStack(info.CreateDate.ToString("yyyy-MM-dd HH:mm:ss"))**  **writer.PushIntoStack(dsItem)**  **return True** |

**运行结果**

**按F5启动程序后，使用协议工具的单元测试功能发起请求测试；**

****