Scut快速开发（2）

Python脚本开发

# 开发环境

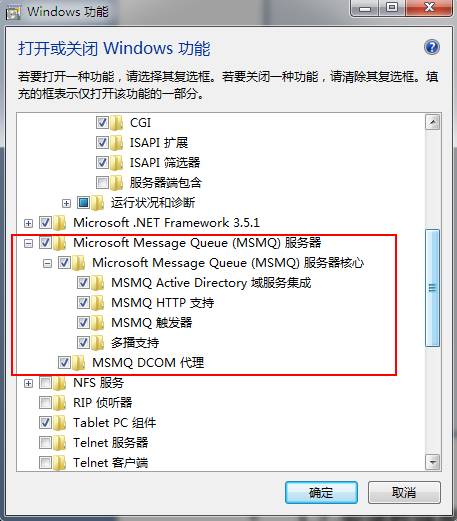
Scut Lib版本：5.2.3.2

**需要安装的软件**

1. IIS和消息队列(MSMQ)

进入控制面板，程序和功能





1. SQL Server2005
2. VS2010开发工具
3. Python2.6（ScutGame官网下载IronPython2.6.1 RC1 for .NET 4.0插件）

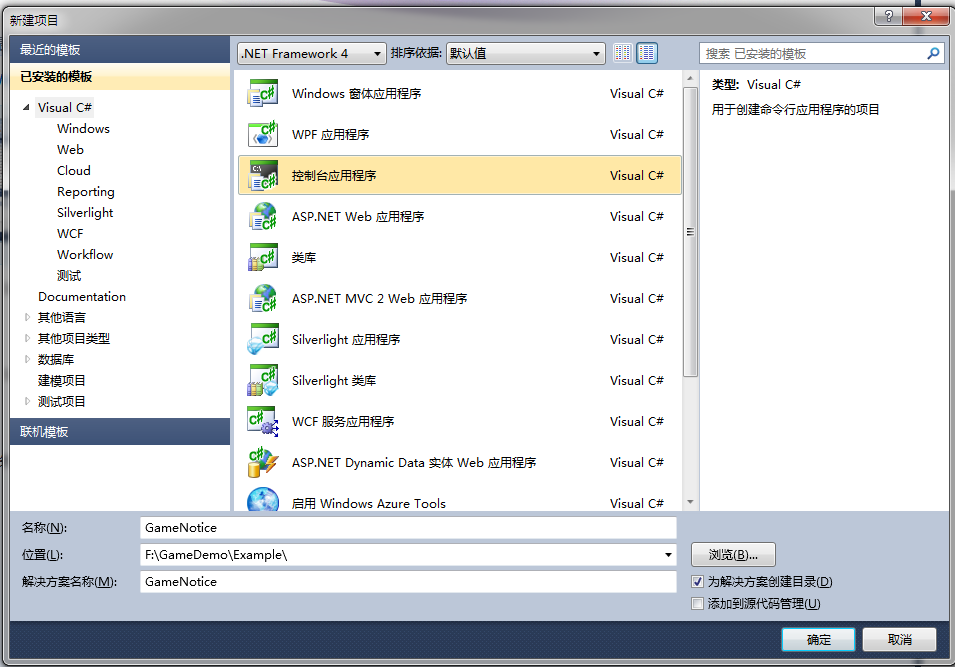
**工具**

1. 协议工具（目录Source\Tools\ContractTools）

# 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** | **Lib\Newtonsoft.Json.dll**  **Lib\NLog.dll**  **Lib\protobuf-net.dll**  **Lib\** **ServiceStack.Common.dll**  **Lib\IronPython.dll**  **Lib\IronPython.Modules.dll**  **Lib\Microsoft.Dynamic.dll**  **Lib\Microsoft.Scripting.dll**  **Lib\ZyGames.Framework.Common.dll**  **Lib\ZyGames.Framework.dll Lib\ZyGames.Framework.Lib.dll**  **Lib\ZyGames.Framework.RPC.dll**  **Lib\ZyGames.Framework.Game.Lang.dll**  **Lib\ZyGames.Framework.Game.dll**  **Lib\** **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GameSocketHost基类，在Main方法中调用

|  |
| --- |
| // Program.cs  static void Main(string[] args)  {  try  {  GameHostApp.Current.Start();  }  catch (Exception ex)  {  Console.WriteLine(ex.Message);  TraceLog.WriteError("HostServer error:{0}", ex);  }  finally  {  Console.WriteLine("Press any key to exit the listener!");  Console.ReadKey();  GameHostApp.Current.Stop();  }  } |

增加GameHostApp类中的OnRequested、OnStartAffer等方法处理代码：

|  |
| --- |
| // GameHostApp.cs  class GameHostApp : GameSocketHost  {  private static GameHostApp instance;  static GameHostApp()  {  instance = new GameHostApp();  }  private GameHostApp()  {  }  public static GameHostApp Current  {  get { return instance; }  }  protected override void OnConnectCompleted(object sender, ConnectionEventArgs e)  {  Console.WriteLine("Client:{0} connect to server.", e.Socket.RemoteEndPoint);  }  protected override void OnRequested(HttpGet httpGet, IGameResponse response)  {  try  {  ActionFactory.Request(httpGet, response, null);  }  catch (Exception ex)  {  Console.WriteLine("{0}", ex.Message);  }  }  protected override void OnStartAffer()  {  try  {  //时º¡À间?间?隔?更¨¹新?库a  int cacheInterval = 600;  GameEnvironment.Start(cacheInterval, () => true);  Console.WriteLine("The server is staring...");  }  catch (Exception ex)  {  TraceLog.WriteError("The server start error:{0}",ex);  }  }  protected override void OnServiceStop()  {  GameEnvironment.Stop();  }  } |

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:**

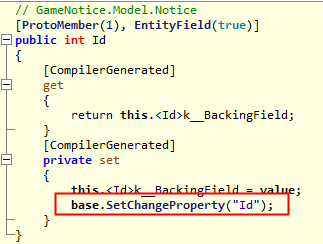
|  |
| --- |
| **<appSettings>**  **<!--服务监听端口-->**  **<add key="Game.Port" value="9001" />**  **<!--Code：游戏代码；ServerId：游戏分区代码-->**  **<add key="Product.Code" value="1"/>**  **<add key="Product.ServerId" value="1"/>**  **<!--Python脚本配置**  **Python\_IsDebug：是否开启调试功能**  **PythonRootPath：脚本路径，相对于程序运行目录**  **-->**  **<add key="Python\_IsDebug" value="true"/>**  **<add key="PythonRootPath" value="..\..\PyScript"/>**  **</appSettings>**  **<connectionStrings>**  **<!--修改DataSource和Pwd-->**  **<add name="GameData" providerName="" connectionString="Data Source=.;Initial Catalog=GameData;Uid=sa;Pwd=123;"/>**  **</connectionStrings>** |

## 实体静态注入配置

用记事本打开GameNotice.csproj文件，在结尾增加如下配置:

|  |
| --- |
| <Project>  ... ...  <UsingTask TaskName="ZyGames.Framework.Common.Build.WeavingEntityTask" AssemblyFile="bin\$(Configuration)\ZyGames.Framework.Common.dll" />  <Target Name="AfterBuild">  <WeavingEntityTask SolutionDir=".\\bin\$(Configuration)" FilePattern="GameNotice.exe" />  </Target>  ... ...  </Project> |

(小提示：使用ILSpy工具反编译可以查看Model的Notice类属性会被修改)



## 创建数据库

使用SQL Server2005建立一个GameData库 ；

## 定义协议接口

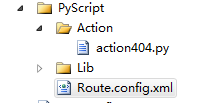
打开协议生成器工具，增加一个“GameNotice”项目方案，接着在增加公告信息（接口编号2001）协议，请求参数和下发参数；如图：



## 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：



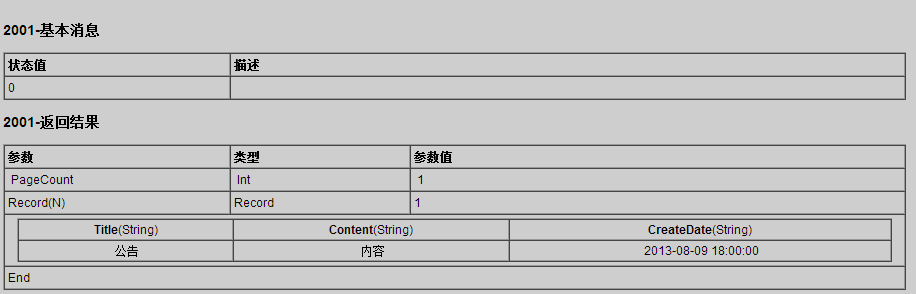
**新增脚本**

**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；**

****