[**Unity中使用多构造函数（转）**](http://www.cnblogs.com/chinadeveloper/p/6184307.html)

如果要实例化的类只有一个构造函数, 则使用方法很简单使用方法如下:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | using (IUnityContainer container = new UnityContainer())  {      UnityConfigurationSection section = (UnityConfigurationSection)ConfigurationManager.GetSection("unity");      section.Configure(container);&nbsp;&nbsp;&nbsp;&nbsp;//...      ILogger logger = container.Resolve<ILogger>("DatabaseLogger");      return logger;  } |

其中配置文件为

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | <?xml version="1.0" encoding="utf-8" ?>  <configuration>    <configSections>      <section name="unity" type="Microsoft.Practices.Unity.Configuration.UnityConfigurationSection,Microsoft.Practices.Unity.Configuration"/>    </configSections>    <unity>      <containers>        <container>          <types>            <type type="Bery.ILogger, UnityStudy" mapTo="Bery.DatabaseLogger, UnityStudy" name="DatabaseLogger">            </type>          </types>        </container>      </containers>    </unity>  </configuration> |

如果DatabaseLogger类中的有两个构造函数, 代码如下

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | public DatabaseLogger()  {  }  public DatabaseLogger(string name)  {  } |

则Unity自动使用参数最多的构造函数进行创建对象, 会抛出以下异常:

|  |  |
| --- | --- |
| 1  2  3 | Microsoft.Practices.Unity.ResolutionFailedException: Resolution of the dependency failed, type = "Bery.ILogger", name = "DatabaseLogger".  Exception occurred while: while resolving.  Exception is: InvalidOperationException - The type String cannot be constructed. You must configure the container to supply this value. |

如果您想让它使用无参的构造函数创建, 则要使用[InjectionConstructor]特性进行修饰无参的构造函数,

|  |  |
| --- | --- |
| 1  2  3  4 | [InjectionConstructor]  public DatabaseLogger()  {  } |

若您想使用带参数的构造函数创建对象, 除了在构造函数上使用[InjectionConstructor]外, 还要在创建时传递参数,代码如下

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | using (IUnityContainer container = new UnityContainer())  {      UnityConfigurationSection section = (UnityConfigurationSection)ConfigurationManager.GetSection("unity");      section.Configure(container);      ILogger logger = container.Resolve<ILogger>("DatabaseLogger",          new ParameterOverrides{          {"name", "logName"}      });      return logger; |