## 主键约定

如果类的属性名为“ID”（不区分大小写）或类名的后面跟有“ID”，则 Code First 会推断该属性是主键。如果主键属性的类型为数值或 GUID，则将其配置为标识列。

publicclass Department

{

  // Primary key

  public int DepartmentID { get;set; }

}

## 关系约定

publicclass Department

{

  // Primary key

  public int DepartmentID { get;set; }

  public string Name { get; set; }

  // Navigationproperty

  public virtual ICollection<Course> Courses { get;set; }

}

publicclass Course

{

  // Primary key

  public int CourseID { get; set; }

  public string Title { get; set; }

  public int Credits { get; set; }

  // Foreign key

  public int DepartmentID { get;set; }

  // Navigationproperties

  public virtual DepartmentDepartment { get; set;}

}

除导航属性外，建议还要包括表示依赖对象的类型的外键属性。

## 复杂类型约定

当 CodeFirst 发现无法推断主键以及未通过数据注释或 Fluent API 注册主键的类时，类型会自动注册为复杂类型。

对于以下类定义，Code First 推断Details 是复杂类型，因为它没有主键。

Public partial class OnsiteCourse : Course

{

  Public OnsiteCourse()

  {

    Details = new Details();

  }

  public Details Details { get;set; }

}

Public class Details

{

  Public System. DateTime Time { get; set; }

  public string Location { get;set; }

  public string Days { get; set; }

}

## 连接字符串约定

### 默认配置

namespaceDemo.EF

{

  public class BloggingContext: DbContext

  {

    Public BloggingContext()

    // C# will callbase class parameterless constructor by default

    {

    }

  }

}

在此示例中，DbContext使用派生上下文类 Demo.EF.BloggingContext 的命名空间限定名称作为数据库名称，并使用 SQL Express 或 LocalDb 为此数据库创建连接字符串。如果同时安装了这两个数据库，将使用 SQL Express。

### 指定数据库名称

如果您尚未在应用程序中进行任何其他配置

namespaceDemo.EF

{

  public class BloggingContext: DbContext

  {

    public BloggingContext()

      : base("BloggingDatabase")

    {

    }

  }

}

### 指定连接字符串

可以选择将连接字符串放入 app.config 或web.config 文件中。例如：

<configuration>

  <connectionStrings>

    <addname="BloggingCompactDatabase"

         providerName="System.Data.SqlServerCe.4.0"

         connectionString="Data Source=Blogging.sdf"/>

  </connectionStrings>

</configuration>

在base中指定连接字符，如不指定则查找与BloggingContext相同的连接字符

publicclass BloggingContext: DbContext

{

  publicBloggingContext()

    : base("BloggingCompactDatabase")

  {

  }

}