12 业务结构对象模型

12.5 配合界面控制层的属性与函数

业务类及其集合类上，还有一些属性和函数可配合界面控制层对界面层上相关的控件进行控制。比如，组件 BarManager（见“10.BarManager 组件”章节）上“增删改查”功能按钮的灰亮控制，及配合 ReadWriteAuthorization 组件（见《“06.ReadWriteAuthorization 组件”章节）实现数据控件的读写控制：

12.5.1Phenix.Business.BusinessBase<T>提供可覆写的属性

属性	说明	备注
AllowEdit	是否允许编辑本对象	一般配合用于控制界面上的编辑功能按钮；但并不代表一定禁止在代码里编辑业务对象，真正用于禁止编辑的属性是业务类的 CanEdit（CanEdit 是权限控制属性）；
AllowDelete	是否允许删除本对象	一般配合用于控制界面上的删除功能按钮；但并不代表一定禁止在代码里删除业务对象，真正用于禁止删除的属性是业务类的 CanDelete（CanDelete 是权限控制属性）；
AllowSet	是否允许设置本对象	是 AllowEdit 和 AllowDelete 属性值的总开关，也就是当 AllowSet 返回 false 时，它们都将返回 false，而且还控制 CanWriteProperty()函数，也会返回 false；

12.5.2Phenix.Business.BusinessBase<T>提供可覆写的函数

```
/// <summary>
/// 允许属性可读
/// </summary>
/// <param name="property">属性信息</param>
/// <returns>属性可读</returns>
public virtual bool AllowReadProperty(Csla.Core.IPropertyInfo property)

/// <summary>
/// 允许属性可写
/// </summary>
/// <param name="property">属性信息</param>
/// <returns>属性可写</returns>
public virtual bool AllowWriteProperty(Csla.Core.IPropertyInfo property)
```

```
    /// <summary>
    /// 允许过程可执行
    /// </summary>
    /// <param name="method">过程信息</param>
    public virtual bool AllowExecuteMethod(Csla.Core.IMemberInfo method)
```

注意，如果在覆写上述重载函数的代码里，业务逻辑上有可能会出现 return true 状况的话，则必须先判断base函数是否会返回false，否则就有可能屏蔽掉基类所提供的缺省规则。以下代码是正确的写法：

```
    /// <summary>
    /// 允许属性可写
    /// </summary>
    /// <param name="property">属性信息</param>
    /// <returns>属性可写</returns>
    public override bool AllowWriteProperty(Csla.Core.IPropertyInfo property)
    {
        if (!base.AllowWriteProperty(property))
          return false;
        return selfAllowWriteProperty; //selfAllowWriteProperty为业务类自身提供的逻辑开关，有可能返回true
    }
```

12.5.3Phenix.Business.BusinessListBase<T, TBusiness>提供可覆写的属性

属性	说明	备注
AllowAddItem	是否允许添加业务对象	一般配合用于控制界面上的新增功能按钮；但并不代表一定禁止在代码里新增业务对象，真正用于禁止新增的的属性是业务类的 CanCreate（CanCreate 是权限控制属性）；
AllowEditItem	是否允许编辑业务对象	一般配合用于控制界面上的编辑功能按钮；但并不代表一定禁止在代码里编辑业务对象，真正用于禁止编辑的属性是业务类的 CanEdit（CanEdit 是权限控制属性）；
AllowDeleteItem	是否允许删除业务对象	一般配合用于控制界面上的删除功能按钮；但并不代表一定禁止在代码里删除业务对象，真正用于禁止删除的属性是业务类的 CanDelete（CanDelete 是权限控制属性）；

12.5.4注意覆写时不能屏蔽掉基类的缺省规则

如果在覆写上述这些属性的代码里，业务逻辑上有可能会出现 return true 状况的话，则必须先判断base属性是否会返回false，否则就有可能屏蔽掉基类所提供的缺省规则。以下代码是正确的写法：

```
/// <summary>
/// 是否允许添加业务对象
/// </summary>
[System.ComponentModel.Browsable(false)]
[System.ComponentModel.DataAnnotations.Display(AutoGenerateField = false)]
public override bool AllowAddItem
{
  get
  {
    if (!base.AllowAddItem)
      return false;
    return selfAllowAddItem; //selfAllowAddItem为业务集合类自身提供的逻辑开关，有可能返回true
  }
}
```