11 业务对象生命周期及其状态

11.2 New 业务对象

本文所述的 New 业务对象，不是指纯粹意义上的“构建对象”，而是指这些 New 出来的业务对象在被持久化的时候，是以新增（insert）记录的方式被提交的。其特征是属性 IsNew = true 且主键字段值是独一无二的（在业务类中为它打上 Phenix.Core.Mapping.FieldAttribute 标签属性 IsPrimaryKey = true）。

因此，Phenix丶是不支持直接（通过 C#的构建对象语法）“new”一个业务对象，然后 Add 到业务集合对象中的编写风格。必须通过以下 Phenix丶所提供的一系列静态方法，来实现真正意义上的新增业务对象。

11.2.1Phenix.Business.BusinessBase<T>提供 New 一个业务对象的函数

```
/// <summary>
/// 新增纯净对象
/// 除WatermarkField外字段不用被填充缺省值
/// </summary>
/// <param name="needFillBusinessCodeFieldValues">是否需要填充业务码字段值</param>
/// <param name="needInitializeNew">是否需要调用OnInitializeNew函数</param>
public static T NewPure(bool needFillBusinessCodeFieldValues, bool needInitializeNew)
```

```
/// <summary>
/// 新增对象
/// 用缺省值填充字段值
/// </summary>
/// <param name="needFillBusinessCodeFieldValues">是否需要填充业务代码字段值</param>
public static T New(bool needFillBusinessCodeFieldValues)
```

11.2.2Phenix.Business.BusinessBase<T>提供复制出一个 New 业务对象的函数

```
/// <summary>
/// 新增对象
/// 按照数据源填充(指定属性，属性名或映射的表字段需一致，忽略对映到本类主键的属性)
/// </summary>
/// <param name="source">数据源</param>
/// <param name="propertyInfos">需匹配的属性信息，当为null、空队列时匹配全部属性</param>
public static T New(IBusinessObject source, params Phenix.Core.Mapping.IPropertyInfo[]
propertyInfos)
```

11.2.3Phenix.Business.BusinessBase<T>提供克隆出一个 New 业务对象的函数

New 业务对象包含被克隆数据源所有层级的从业务对象的克隆版，即所谓的深度克隆，这些从业务对象亦是 New 业务对象，同时与父业务对象之间的关系亦被重新自动勾连。

```
/// <summary>
/// 新增对象
/// 主键(包括Details)重新生成
/// </summary>
/// <param name="cloneSource">Clone数据源</param>
public static T New(T cloneSource)
{
  return cloneSource.Clone(true);
}
```

上述函数实质是调用了 Clone 数据源的 Clone()函数：

```
/// <summary>
/// 克隆
/// </summary>
/// <param name="isNew">全新的</param>
public T Clone(bool isNew)
```

只要 isNew 参数等于 true，则返回的就是一个全新的 New 业务对象。

使用本功能必须注意的是，在调用 Clone()函数之前的 Clone 数据源，其从业务（集合）对象是不是已经被 Fetch 到本地了。

比如一般是采用如下方式获取到从业务（集合）对象，并通过属性展现出来的：

```
/// <summary>
/// 程序集类信息
/// </summary>
public AssemblyClassInfoList AssemblyClassInfos
{
  get { return GetCompositionDetail<AssemblyClassInfoList, AssemblyClassInfo>(); }
}
```

如果没有显式调用 AssemblyClassInfos 属性，Phenix 丫是不会主动将这个从业务（集合）对象 Fetch 到本地的。

所以，要是希望克隆出的业务对象里包含到这些从业务（集合）对象的话，必须先试着 Fetch 它们一次，以免出现软性故障（时有时无、规律难寻），增加调试难度。比如，一种做法是，我们可以在业务类中覆写 GetClone()函数：

```
protected override object GetClone()
{
   // 尝试Fetch从业务对象
   AssemblyClassInfoList assemblyClassInfos = AssemblyClassInfos;
   // 正式克隆
  return base.GetClone();
}
```

11.2.4Phenix.Business.BusinessBase<T>提供业务对象自己置换成一个 New 业务对象的函数

所谓置换，就是业务对象将自己摇身一变为全新的业务对象，它的内存空间和引用指针其实并没有发生改变，但是它与原来的业务对象之间，从逻辑上来讲已完全不搭界了。

如果被置换的业务对象包含有从业务对象，亦将被迭代置换，也就是说，这些从业务对象亦是 New 业务对象，同时与父业务对象之间的关系亦被重新自动勾连。

```
/// <summary>
/// 标为 IsNew = true
/// 主键(包括Details)重新生成
/// </summary>
protected override void MarkNew()
```

11.2.5Phenix.Business.BusinessListBase<T, TBusiness>提供 New 一个业务对象并 Add 到业务对象集合中的函数

```
/// <summary>
/// 新增
/// </summary>
/// <param name="index">索引</param>
/// <param name="cloneSource">Clone数据源</param>
public TBusiness AddNew(int index, TBusiness cloneSource)

/// <summary>
/// 新增
```

```
    /// </summary>
    /// <param name="index">索引</param>
    /// <param name="source">数据源</param>
    /// <param name="propertyInfos">需匹配的属性信息，当为null、空队列时匹配全部属性</param>
    public TBusiness AddNew(int index, IBusinessObject source, params Phenix.Core.Mapping.IPropertyInfo[]
propertyInfos)
```

11.2.6Phenix.Business.BusinessListBase<T, TBusiness>提供从 DataTable 数据源 New 一组业务对象的函数

```
    /// <summary>
    /// 新增业务对象集合
    /// </summary>
    /// <param name="source">数据源</param>
    /// <param name="propertyInfos">数据属性信息队列，顺序与数据源columnIndex一致，当为null、空队列时
按source列名与业务类属性名匹配条件进行数据填充</param>
    public static T New(DataTable source, params Phenix.Core.Mapping.IPropertyInfo[] propertyInfos)
```

一般应用场景是用于从应用系统外部数据源（比如 Execl 清单）中导入业务数据。

以下是示例如何通过弹出对话框，由用户选取 Execl 表单（参数 sheetName 是缺省 execl 文件名）将该 Execl 表单数据转换为 DataTable 的代码：

```
    DataTable dataTable = Phenix.Windows.Helper.ExcelHelper.Import(sheetName);
```