

文章

[water huang](#) · 九月 21 阅读大约需 4 分钟

## 对 %XML.PropertyParameters类的探索

日常工作中，我们使用xml 应该还是比较多的，早期的ensemble或者cache，对JSON的支持不是很好，但是对xml支持比

较好。因

此早期的接口方式

中，webservice+xml的方式很常用。

而对象导出为xml最简单的方式就是类继承

%XML.Adaptor。继承该类后，就可以使用很多xml的特性，以满足各种需求。虽然看不到对象转为xml的具体实现，但是配置了参数，就可以实现更多的效果。那么想自定义一些参数呢？如果有人打开过它的定义，可以看到类上面有个 PropertyClass =

%XML.PropertyParameters”，那么我们其实在需要xml参数的同时，又要自定义参数，就可以自己写个属性类，继承于%XML.PropertyParameters。

这里，我姑且定义一个类，用于说明使用场景。

```
Class TEST.PropertyParameters Extends %XML.PropertyParameters [ ProcedureBlock ]
```

```
{  
  
    /// 敏感信息过滤  
    Parameter SENSITIVEINFOCTRL As STRING [ Constraint = ",*", Flags = ENUM ];  
  
    /// 可以查表  
    Parameter DICLookUp As STRING [ Constraint = ",Y,N", Flags = ENUM ];  
  
    /// 主键  
    Parameter KEY As STRING [ Constraint = ",Y,Y1,Y2,Y3", Flags = ENUM ];  
  
    /// 数据格式 8位日期，6位时间，15位日期时间，整数,小数  
    Parameter DataFm As STRING [ Constraint = ",D8,D10,T6,T8,DT,DT15,N,F,SF,YW", Flags = ENUM ];  
  
    /// 标准化名称  
    Parameter STDName As STRING;  
  
    ///字典数据  
    Parameter DICCTRL [ Constraint = ",SEX,ABOTYPE", Flags = ENUM ];  
  
}
```

现在有个应用场景是

把业务系统的数据做值域转换后写入第三方的数据库。常用的方式可能是写query，然后通过etl工具，比如kettle。这里我们使用的方案是HIS把数据发给ensemble平台，然后ensemble平台处理数据后写入数据库。

定义一个类；

```
///患者信息
```

```
Class TEST.HZXX Extends (%RegisteredObject, %XML.Adaptor) [ Inheritance =  
right, Not ProcedureBlock, PropertyClass = TEST.PropertyParameters ]
```

```
{
```

```
/// 性别代码(字符-GB/T2261.1-2003生理性别代码表-)
```

```
Property XBDM As %String(DICCTRL = "SEX", DICLookUp = "Y", MAXLEN = 1, TRUNCATE = 1,  
XMLNAME = "XB",STDName="XMMS");
```

```
///性别描述
```

```
Property XMMS As %String(MAXLEN = 25, TRUNCATE = 1, XMLNAME = "XMMS");
```

```
/// 医疗机构代码(字符串-医疗机构在区域卫生信息平台的唯一识别码-)
```

```
Property YLJGDM As %String(KEY = "Y1", MAXLEN = 22, TRUNCATE = 1, XMLNAME = "YLJGDM")  
[ Required ];
```

```
/// 个人标识类型(字符串-复合主键；患者识别码，与患者各种就诊信息记录进行关联-)
```

```
Property GRBSLX As %String(KEY = "Y3", MAXLEN = 16, TRUNCATE = 1, XMLNAME = "GRBSLX") [ Required ];
```

```
/// 个人标识号(字符串--复合主键；患者识别码，与患者各种就诊信息记录进行关联)
```

```
Property GRBSH As %String(KEY = "Y2", MAXLEN = 36, TRUNCATE = 1, XMLNAME = "GRBSH")  
Parameter XMLIGNOREINVALIDTAG = 1;
```

```
Parameter XMLIGNORENULL = 1;
```

```
}
```

这里用到了KEY,DICCTRL,DICLookUp,STDName.而这些都是自定义的。

假设 HIS发出的数据是xml数据。那么在平台解析为对象后，可以把一个一个的对象做值域转换后，再转为sql，然后操作数据库。

我们用于

值域转换的方法的

实现，就不详细说，主要说下逻辑过

程：1.遍历类定义的属性。如果属性的参数

DICLookUp="Y",表示它是需要做值域对照的。然后下一步检查 DICCTRL的值，如果不为空，说明它可以查询值域对照表（值域对照表就是HIS代码，HIS名称，第三方代码，第三方名称

这样的对

照好了的数据。）

了，如果查询到了对照关系，首先给

这个属性赋值为第三方的代码。下一步检查

STDName是否有配置（需要配置为该类中的属性名），如果配置了，那么就给他赋值为第三方的描述，巧妙的是它可以赋值为属性本身，也即是这样的场景：如果该属性要求的是标准的说明，那么我们可以取值为代码，STDName配置为属性本身，这样就可以完成这样的需求。至此值域转换完成。下一步就是转为sql。可能有两个情况：1.数据不存在，那么需要插入语句，2.如果数据存在，那么需要转为更新语句，更新语句需要只更新这条数据，那么KEY的作用就显示出来了，它是作为主键，也就是更新的条件。3.通过上面说的内容，其实对于oracle数据库，日期时间的数据，是需要使用函数转换下的，因此可给属性配置数据格式，然后转换为sql语句的时候，可以通过这个配置转出正确的sql语句。

这个方案可以把值域对照和数据取值区分开。使用上比在取值的时候对照更灵活。如果大家有更好的建议，欢迎留言

[#API #Caché #Ensemble](#)

---

源

URL:

<https://cn.community.intersystems.com/post/%E5%AF%B9-xmlpropertyparameters%E7%B1%BB%E7%9A%84%E6%8E%A2%E7%B4%A2>