

文章  
姚鑫 · 六月 8, 2021 阅读大约需分钟

## 第一章 InterSystems XML工具简介

### 第一章 InterSystems XML工具简介

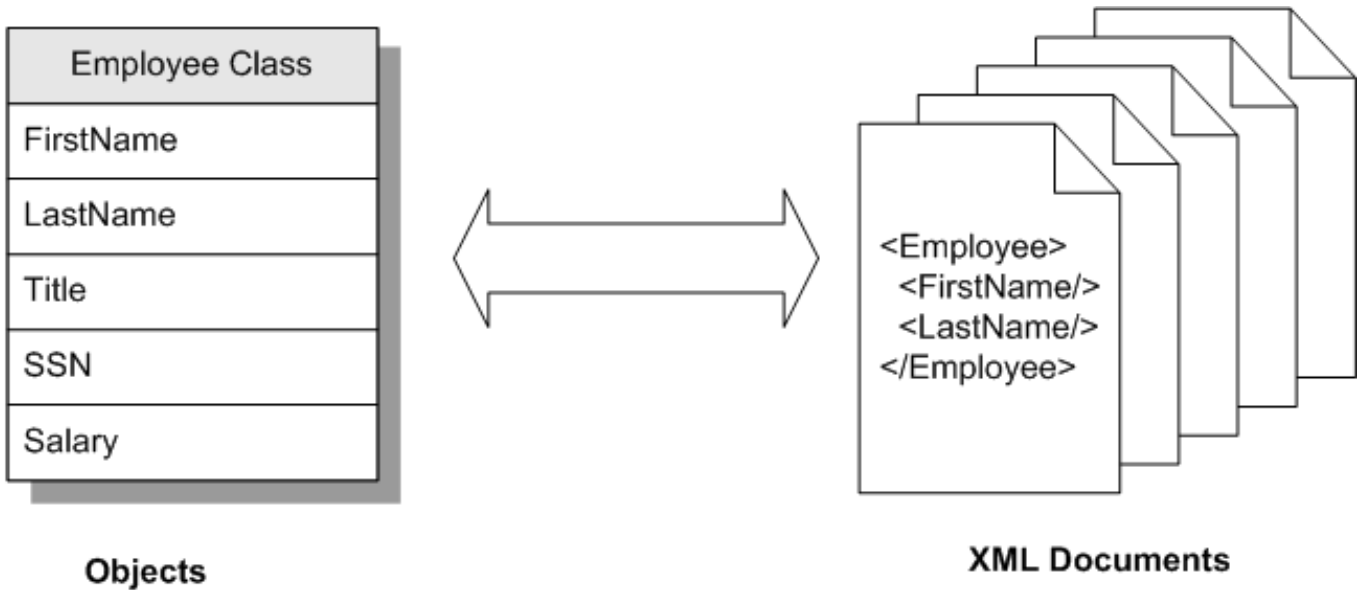
介绍了如何使用 IRIS XML工具。

#### InterSystems

IRIS为XML处理带来了对象的力量--可以使用对象作为XML文档的直接表示,反之亦然。由于InterSystems IRIS包括本机对象数据库,因此可以将此类对象直接用于数据库。此外,InterSystems IRIS提供了用于处理XML文档和DOM(文档对象模型)的工具,即使它们与任何InterSystems IRIS类无关。

### 用XML表示对象数据

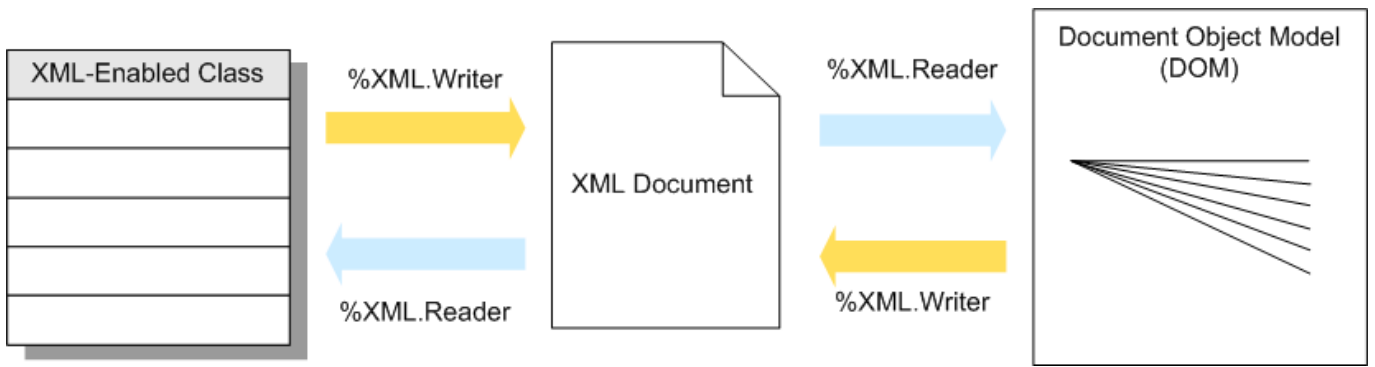
有些InterSystems IRIS XML工具主要用于支持XML的类。要为类启用XML,需将%XML.Adaptor添加到其超类列表中。%XML.Adaptor类使能够将该类的实例表示为XML文档。可以添加类参数和属性来微调影响。



对于启用了XML的类,数据可以采用以下所有形式:

- 包含在类实例中。根据类的不同,还可以将数据保存在磁盘中,在磁盘中数据可以像其他持久类一样以所有相同的方式使用。
- 包含在XML文档中,可以是文件、流或其文档。
- 包含在DOM(文档对象模型)中。

图概述了用于在这些表单之间转换数据的工具:



`%XML.Writer`类使能够创建XML文档。输出目的地通常是文件或流。确定要包括在输出中的对象，系统根据在类定义中建立的规则生输出。

`%XML.Reader`类使能够将合适的XML文档导入到类实例中。源通常是文件或流。要使用此类，需指定类名和XML文档中包含的元素之间的关联。给定的元素必须具有相应类所需结构。然后您逐个节点地阅读文档。这样做时，系统会创建该类的内存中实例，其中包含在XML文档中找到的数据。

DOM也是处理XML文档的有用方法。可以使用`%XML.Reader`类读取XML文档并创建表示它的DOM。在此表示中，DOM是一系列节点，可以根据在它们之间导航。具体来说，将创建`%XML.Document`的一个实例，该实例表示文档本身并包含节点。然后使用`%XML.Node`检查和操节点。如果需可以使用`%XML.Writer`重新编写XML文档。

InterSystems IRIS XML工具提供了许多方法来访问和修XML文档和DOM中的数据。

## 创建任意XML

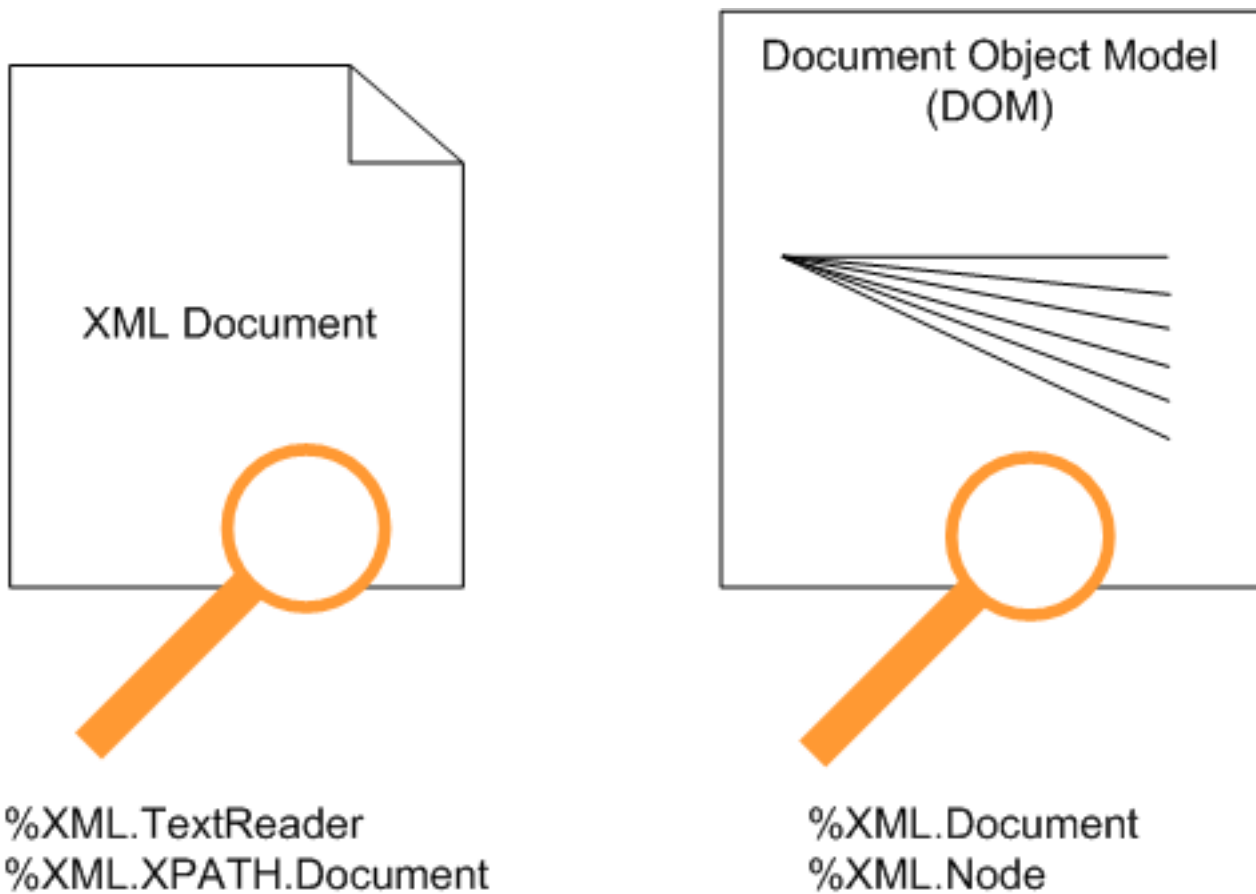
还可以使用InterSystems IRIS XML工具创建和使用任意XML-即不映射到任何InterSystems IRIS类的XML。要创建任意XML文档，请使用`%XML.Writer`。该类提供了用于添加元素、添加属性添加命名空间声明等方法。

要创建任意DOM，请使用`%XML.Document`。该类提供了一个类方法，该方法具有单个空节点的DOM。然后根据需使用该类的实例方法添加节点。

或者使用`%XML.Reader`读取任意XML文档，然后从该文档创建DOM。

## 访问数据

InterSystems IRIS XML工具提供了几种访问XML格式数据的方法。图显示了：



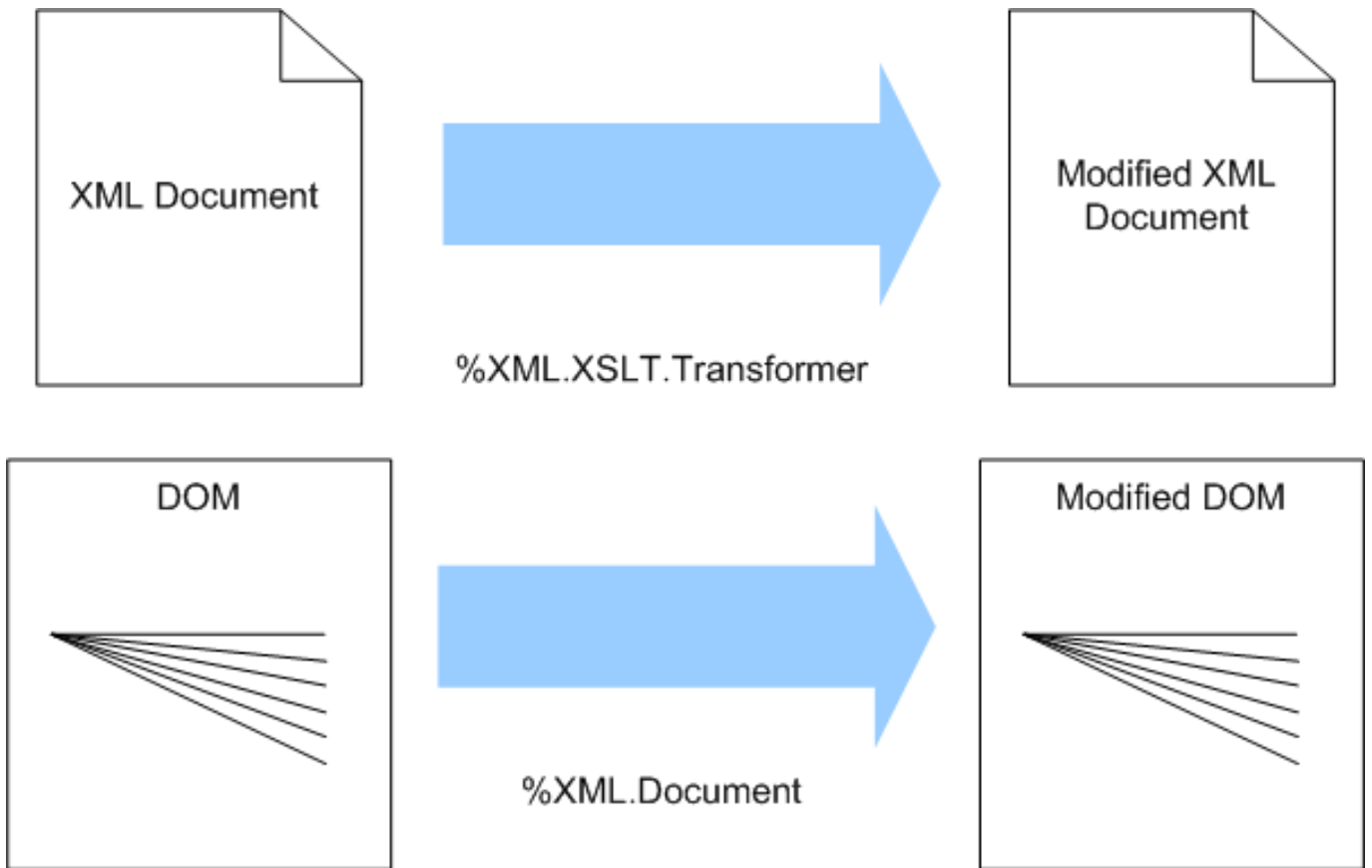
对于任何格式良好的XML文档，都可以使用以下类来处理该文档中的数据：

- `%XML.TextReader`-可以使用它逐个节点地读取和解析文档。
- `%XML.XPATH.Document`-可以使用它来获取数据，方法是使用引用文档中特定节点的XPath表达式。

在InterSystems IRIS中，DOM是`%XML.Document`的实例。该实例表示文档本身并包含节点。可以使用该类的属性和方法从DOM中检索值。可以使用`%XML.Node`检查和操作节点。

## 修ML

InterSystems IRIS XML工具还提供了修ML格式数据的方法。图显示了修：



对于XML文档,可以使用%XML.XSLT.Transformer中的类方法执行XSLT转换并获得文档的副本。

对于DOM,可以使用%XML.Document的方法修改DOM。例如,可以添加或删除元素或属性。

## SAX解析器

InterSystems IRIS XML工具使用InterSystems IRIS SAX(Simple API For XML)解析器。这是一个内置的SAX XML验证解析器,使用标准Xerces库。SAX是一个解析引擎,它提供完整的XML验证和文档解析。InterSystems IRIS SAX使用高级进程内调入机制与InterSystems IRIS进程通信。使用此解析器,可以使用内置的InterSystems IRIS XML支持或通过InterSystems IRIS中提供您自己的自定义SAX接口类来处理XML文档。

对于特殊应用程序,可以创建自定义实现解析器和内容处理程序。

可以使用行业标准的XMLDTD或模式验证来验证任何传入的XML,并且可以指定要解析的XML项。

## 其他XML工具

InterSystems IRIS XML支持包括以下附加工具:

- XML架构向导读取XML架构文档,并生成组支持XML的类,这些类与架构中定义的类型相对应。可以指定一个包来包含类,以及控制类定义详细信息的种选项。
- %XML.Schema类使能够从一组启用了XML的类生成XML架构。
- %XML.Namespaces类使能够检查XML命名空间以及其中的类,以查找InterSystems IRIS命名空间。
- %XML.Security.EncryptedData类和其他类使能够加密XML文档以及解密加密文档。
- %XML.Security.Signature类和其他类使能够对XML文档进行数字签名,以及验证数字签名。

## 使用XML工具时的 注意事项

在使用任何类型的XML工具时，至少有三个一般要点需要考虑：

- 任何XML文档都有字符编码
- 将XML文档映射到类(文字或SOAP编码)有不同的方法
- 应该知道SAX解析器的默认行为

## 输入输出的字符编码

导出XML文档时，可以指定要使用的字符编码；否则，InterSystems IRIS会根据目标选择编码：

- 如果输出目标是文件或二进制流，则默认值为“UTF-8”。
- 如果输出目标是字符串或字符流，则默认为“UTF-16”。

对于InterSystems IRIS读取的任何XML文档，文档的XML声明应该指示该文件的字符编码，并且文档应该按照声明的方式进行编码。例如：

```
<?xml version="1.0" encoding="UTF-16"?>
```

但是，如文档中未声明字符编码，InterSystems IRIS将假定：

- 如果输出目标是文件或二进制流，则默认值为“UTF-8”。
- 如果输出目标是字符串或字符流，则默认为“UTF-16”。

## 选择文档格式

使用XML文档时，必须知道将文档映射到InterSystems

IRIS类时要使用的格式。同样，在创建XML文档时，需指定编写文档时要使用的文档格式。XML文档格式如：

- 文字表示文档是对象实例的文字副本。在大多数情况，即使在使用SOAP时，也使用文字格式。除非另有说明，否则文档中的示例均使用文字格式。
- 编码的意思是按照SOAP 1.1标准或SOAP 1.2标准中描述的编码。SOAP1.1和SOAP1.2的细节略有不同。

以下小节显示了这些文档格式之间的差异。

## 文字格式

```
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <Person>
    <Name>Klingman,Julie G.</Name>
    <DOB>1946-07-21</DOB>
    <GroupID>W897</GroupID>
    <Address>
      <City>Bensonhurst</City>
      <Zip>60302</Zip>
    </Address>
    <Doctors>
      <DoctorClass>
        <Name>Jung,Kirsten K.</Name>
      </DoctorClass>
    </DoctorClass>
  </Person>
</Root>
```

```

        <Name>Xiang, Charles R.</Name>
    </DoctorClass>
    <DoctorClass>
        <Name>Frith, Terry R.</Name>
    </DoctorClass>
</Doctors>
</Person>
</Root>

```

## 编码格式

相比之下，下面的示例以编码格式显示相同的数据：

```

<?xml version="1.0" encoding="UTF-8"?>
<Root xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
...
    <DoctorClass id="id2" xsi:type="DoctorClass">
        <Name>Jung, Kirsten K.</Name>
    </DoctorClass>
...
    <DoctorClass id="id3" xsi:type="DoctorClass">
        <Name>Quixote, Umberto D.</Name>
    </DoctorClass>
...
    <DoctorClass id="id8" xsi:type="DoctorClass">
        <Name>Chadwick, Mark L.</Name>
    </DoctorClass>
...
    <Person>
        <Name>Klingman, Julie G.</Name>
        <DOB>1946-07-21</DOB>
        <GroupID>W897</GroupID>
        <Address href="#id17" />
        <Doctors SOAP-ENC:arrayType="DoctorClass[3]">
            <DoctorClass href="#id8" />
            <DoctorClass href="#id2" />
            <DoctorClass href="#id3" />
        </Doctors>
    </Person>
    <AddressClass id="id17" xsi:type="s_AddressClass">
        <City>Bensonhurst</City>
        <Zip>60302</Zip>
    </AddressClass>
...
</Root>

```

请注意编码版本中的以下差异：

- 输出的根元素包括SOAP编码命名空间和其他标准命名空间的声明。
- 本文档包括同一级别的人员、地址和医生元素。Address和Doctor元素列出了引用它们的Person元素使用的唯一ID。每个对象值属性都是这样处理的。
- 顶级Address和Doctor元素的名称与各自类的名称相同，而不是与引用它们的属性名称相同。
- 编码格式不包括任何属性。GroupID属性映射为Person类中的属性。在文字格式中，此属性被影射为特性。

- 。但是,在编码版本中,属性影为元素。
- 对集合的处理方式不同。例如,列表元素具有属性NC:arrayType。
- 每个元素都有一个xsi:type属性值。

注意:对于SOAP1.2,编码版本略有不同。要轻松区分版本,请检查SOAP编码命名空间的声明:

- 对于SOAP1.1,SOAP编码命名空间为"http://schemas.xmlsoap.org/soap/encoding/"
- 对于SOAP1.2,SOAP编码命名空间为"http://schemas.xmlsoap.org/wsdl/soap12/"

除非解析器可以使用这些其他模式,否则验证将失败。特别是对于WSDL文档,有时需加载所有模式并编辑主模式以使用正确的位置。

它尝试解析所有实体包括所有外部实体其他XML解析器也会这样做。)。这一过程可能很耗时,具体取决于它们所在的位置。具体来说,Xerces使用网络访问器来解析一些URL,并且实现使用阻塞I/O。因此,不会超时,网络获取可能会在错误条件挂起,这在实践中很少见。

此外,Xerces不支持https;也就是说,它不能解析位于https位置的实体

如果需要创建自定义实体解析器,也可以禁用实体;

## IRIS支持的标准

IRIS XML支持遵循以下标准:

- XML 1.0 (<https://www.w3.org/TR/REC-xml/>)
- Namespaces in XML 1.0 (<https://www.w3.org/TR/REC-xml-names/>)
- XML Schema 1.0 (<https://www.w3.org/TR/xmlschema-0/>, <https://www.w3.org/TR/xmlschema-1/>, <https://www.w3.org/TR/xmlschema-2/>)
- XPath 1.0 as specified by <https://www.w3.org/TR/xpath>
- SOAP 1.1标准第5节指定的SOAP 1.1编码。
- SOAP1.2编码,如第3节第2部分: Adjuncts ( <https://www.w3.org/TR/soap12-part2/>) of the SOAP 1.2 standard.
- XML Canonicalization Version 1.0 (also known as inclusive canonicalization), as specified by <https://www.w3.org/TR/xml-c14n>.
- XML Exclusive Canonicalization Version 1.0 as specified by <https://www.w3.org/TR/xml-exc-c14n/>, including the InclusiveNamespaces PrefixList feature (<https://www.w3.org/TR/xml-exc-c14n/#def-InclusiveNamespaces-PrefixList>)
- XML Encryption (<https://www.w3.org/TR/xmlenc-core/>)

InterSystems IRIS支持使用RSA-

OAEP或RSA-1.5进行密钥加密, 并支持使用AES-128、AES-192或AES-256对邮件正文进行数据加密。

- XML Signature using Exclusive XML Canonicalization and RSA SHA-1 (<https://www.w3.org/TR/xmlsig-core/>)

InterSystems IRIS提供两个XSLT处理器:

- Xalan处理器支持XSLT 1.0,
- Saxon处理器支持XSLT 2.0,

**注意:** InterSystems IRIS不支持在一个元素中有多个名称相同的属性每个属性于不同的名称空间中。

[#XML](#) [#Caché](#) [#InterSystems IRIS](#)

源 URL: <https://cn.community.intersystems.com/post/%E7%AC%AC%E4%B8%80%E7%AB%A0-intersystems-xml%E5%B7%A5%E5%85%B7%E7%AE%80%E4%BB%8B>