文章
姚 鑫 · 六月 17, 2021 阅读大约需 5 分钟

第十章 XML元素和属性

第十章 XML元素和属性

检查必需的元素和属性

默认情况下，next()方法不检查是否存在与标记为必需的属性相对应的元素和属性。要使读取器检查此类元素和属性是否存在，请在调用Next()之前将读取器的CheckRequired属性设置为1。出于兼容性原因，此属性的默认值为0。

如果将CheckRequired设置为1，并且调用next()，而导入的XML缺少必需的元素或属性，则next()方法会将sc参数设置为错误代码。例如：

```
SAMPLES>set next= reader.Next(.object,.status)

SAMPLES>w next
0
SAMPLES>d $system.Status.DisplayError(status)

ERROR #6318: Property required in XML document: ReqProp
```

处理意外的元素和属性

由于源XML文档可能包含意外的元素和属性，因此%XML.Adaptor提供参数来指定导入此类文档时的反应方式。

控制如何导入空元素和属性

为对象启用XML时，需要指定将空值和空字符串投影到XML的方式

其中一个选项是在支持XML的类中将XMLIGNORENULL设置为等于“ Runtime ”(不区分大小写)。在这种情况下，当使用%XML. IRIS对象时， IRIS使用读取器的IgnoreNull属性的值来确定如何处理空元素或属性，如下所示：

- 如果读取器的IgnoreNull属性为0(默认值)，并且元素或属性为空，则相应的属性设置为等于$char(0)
- 如果读取器的IgnoreNull属性为1，并且元素或属性为空，则不会设置相应的属性，因此等于“ ”

读取器的IgnoreNull属性无效，除非XMLIGNORENULL在启用XML的类中为“ Runtime ”；XMLIGNORENULL的其他可能值为0(默认值)、1和“ INPUTONLY ”；

示例：IgnoreNull为0(默认值)

```
Class EmptyStrings.Import Extends (%Persistent, %XML.Adaptor)
{

Parameter XMLNAME="Test";
```

```
///Reader will set IgnoreNull property
Parameter XMLIGNORENULL = "RUNTIME";

Property PropertyA As %String;

Property PropertyB As %String;

Property PropertyC As %String;

Property PropertyD As %String(XMLPROJECTION = "ATTRIBUTE");

Property PropertyE As %String(XMLPROJECTION = "ATTRIBUTE");
}
```

以下XML文件：

```
<?xml version="1.0" encoding="UTF-8"?>
<Test PropertyD="">
  <PropertyA></PropertyA>
  <PropertyB xsi:nil="true"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
</Test>
```

如果创建%XML.Reader实例，并使用它将此文件导入到上一个类中，则会得到以下结果：
- PropertyA和PropertyD等于$char(0)。
- 所有其他属性都等于“ ”。

例如，如果编写了一个例程来检查每个属性、检查其值并写入输出，则可能会得到类似以下内容：

```
PropertyA is $char(0)
PropertyB is null
PropertyC is null
PropertyD is $char(0)
PropertyE is null
```

示例：IgnoreNull为1

在前面示例的变体中，我们将读取器的IgnoreNull属性设置为1。在这种情况下，所有属性都等于“ ”。

例如，如果编写了一个例程来检查每个属性、检查其值并写入输出，则可能会得到类似以下内容：

```
PropertyA is null
PropertyB is null
PropertyC is null
PropertyD is null
PropertyE is null
```

跳过输入文档的较早部分

XML文档由一组节点组成；根节点编号为0，其中的第一个元素编号为1，依此类推。可以指定开始读取的节点；这对于大型文档特别有用。为此，请设置读取器的Node属性。对于该值，请指定一个整数。

下面是一个示例方法：

```
ClassMethod DemoSkippingAhead(nodenumber As %Integer = 0)
{
    Set cls="GXML.Person"
    Set filename="c:\GXML.Person.xml"
    Set element="Person"

    // ??%XML.Reader???
    Set reader = ##class(%XML.Reader).%New()

    // ??????
    Set status = reader.OpenFile(filename)
    If $$$ISERR(status) {Do $System.Status.DisplayError(status)}

    // ????XML??????
    Do reader.Correlate(element,cls)

    //????????
    Set reader.Node=nodenumber

    // ?XML??????
    While (reader.Next(.object,.status)) {
        Write "Node number "_reader.Node_" contains "_object.Name,!
    }
}
```

此方法假定有特定的输入文件、类名和元素名。默认情况下，此方法从文件开头开始。例如：

```
GXML>d ##class(Readers.ReadFile).DemoSkippingAhead()
Node number 3 contains Emerson,Chad I.
Node number 30 contains O'Rielly,Patricia L.
Node number 63 contains Hanson,Brendan T.
Node number 120 contains Orwell,Tara H.
Node number 195 contains Gold,Elvis O.
Node number 234 contains Klein,Brenda U.
Node number 252 contains Yezek,Kristen Q.
Node number 327 contains Quine,Angelo B.
Node number 378 contains Vivaldi,Milhouse J.
Node number 453 contains Yezek,Vincent D.
Node number 471 contains Xander,Juanita D.
Node number 522 contains Winters,Kyra R.
Node number 555 contains Woo,Michelle J.
Node number 636 contains Ihringer,Yan A.
Node number 654 contains West,Hannah N.
Node number 729 contains Xiang,Bob G.
Node number 762 contains Ximines,Howard H.
Node number 789 contains Quixote,Jocelyn P.
Node number 864 contains Hills,Charles E.
Node number 897 contains Evans,Milhouse R.
```

相比之下，我们可以跳过以下步骤：

```
GXML>d ##class(Readers.ReadFile).DemoSkippingAhead(700)
Node number 729 contains Xiang,Bob G.
Node number 762 contains Ximines,Howard H.
Node number 789 contains Quixote,Jocelyn P.
Node number 864 contains Hills,Charles E.
Node number 897 contains Evans,Milhouse R.
```

其他有用的方法

有时，可能需要使用%XMLReader的以下其他方法
- 如果需要从头开始读取XML源文档，请使用Rewind()方法。此方法清除所有关联。
- 如果要显式关闭和清理导入处理程序，请使用Close()方法。导入处理程序会自动清除；包含此方法是为了向后兼容。

Reader属性

可以设置%XML.Reader的以下属性。以控制方法的整体行为：

- 使用UsePPGHandler属性指定%XML.Reader的实例在分析文档时是否使用进程私有全局变量。如果此属性为true，则实例使用进程私有全局变量。如果此属性为false，则实例使用内存。如果未设置此属性(或等于空字符串)，则实例使用默认值，通常为内存。

使用Format属性指定XML文档的整体格式。指定下列值之一：

- “ literal ”，默认值，在本章的大多数示例中都使用了它。
- “ encoded ”，按照SOAP 1.1标准中的描述进行编码。
- “ encoded12 ”，按照SOAP 1.2标准中的描述进行编码。

注意，可以在OpenFile()、OpenStream()、OpenString()和OpenURL()方法中重写Format属性。

除非使用关联()和Next()，否则此属性无效。

- 使用Summary属性强制读取器仅导入启用XML的对象的摘要字段。如将对象投影到XML中所述，对象的摘要由其XMLSUMMARY类参数指定，可以将其指定为逗号分隔的属性列表。
- 使用IgnoreSAXWarnings属性指定读取器是否应该报告SAX解析器发出的警告。

%XML.Reader还提供可用于检查正在阅读的文档的属性：

- Document属性包含%XML.Document实例，该实例代表正在阅读的整个已分析文档。
- Node属性是一个字符串，表示XML文档的当前节点。请注意，0表示文档，即根元素的父元素。

#Caché #InterSystems IRIS

源 URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E5%8D%81%E7%AB%A0-xml%E5%85%83%E7%B4%A0%E5%92%8C%E5%B1%9E%E6%80%A7