

文章

[姚鑫](#) · 六月 13, 2021 阅读大约需 6 分钟

第六章 控制名称空间的使用

第六章 控制名称空间的使用

控制名称空间的使用

如将对象投射到XML中所述，可以将类分配给名称空间，以便相应的XML元素属于该名称空间，还可以控制类的属性是否也属于该名称空间。

将类中的对象导出为XML时，%XML.Write提供其他选项，例如指定元素是否为其父级的本地元素。本节包括以下主题：

- 默认情况下，%XML.Writer如何处理命名空间
- 如何指定本地元素是否合格
- 如何指定元素是否为其父元素的本地元素
- 如何指定属性是否合格
- 命名空间分配方式的摘要

注意：在InterSystems IRIS XML支持中，可以按类指定名称空间。通常，每个类都有自己的命名空间声明；但是，通常只需要一个或少量的命名空间。还可以在逐个类的基础上指定相关信息(而不是以某种全局方式)。这包括控制元素是否为其父元素的本地元素以及子元素是否合格的设置。为简单起见，建议使用一致的方法。

名称空间的默认处理

若要将启用XML的类分配给命名空间，请设置该类的Namespace参数，如将对象投影到XML中所述。在%XML.Writer会自动插入命名空间声明，生成命名空间前缀，并在适当的地方应用前缀。例如，以下类定义：

```
Class GXML.Objects.WithNamespaces.Person Extends (%Persistent, %Populate, %XML.Adaptor)
{
    Parameter NAMESPACE = "http://www.person.com";
    Property Name As %Name [ Required ];
    Property DOB As %Date(FORMAT = 5, MAXVAL = "+$h") [ Required ];
    Property GroupID As %Integer(MAXVAL=10, MINVAL=1, XMLPROJECTION="ATTRIBUTE");
}
```

```
<Person xmlns="http://www.person.com" GroupID="4">
  <Name>Uberoth, Amanda Q.</Name>
  <DOB>1952-01-13</DOB>
</Person>
```

请注意以下事项：

- 名称空间声明被添加到每<Person>元素。

- 默认情况下，<Person>元素的局部元素(<Name>和<DOB>)是限定的。该名称空间被添加为默认名称空间，因此应用于这些元素。
- <Person>元素的属性(GroupID)默认是无限定的。这个属性没有前缀，因此被认为是无限定的。
- 这里显示的前缀是自动生成的。(请记住，当对象分配给名称空间时，只指定名称空间，而不是前缀。)
- 此输出不会在写入器中设置任何与名称空间相关的属性，也不会写入器中使用任何与名称空间相关的方法。

命名空间分配的上下文效应

为支持xml的对象分配的名称空间取决于该对象是在顶层导出还是作为另一个对象的属性导出。

一个名为Address的类。

假设使用NAMESPACE参数将Address类分配给名称空间“http://www.address.org”。

如果你直接导出Address类的一个对象，你可能会收到如下输出：

```
<Address xmlns="http://www.address.org">
  <Street>8280 Main Avenue</Street>
  <City>Washington</City>
  <State>VT</State>
  <Zip>15355</Zip>
</Address>
```

注意，<Address> 元素及其所有元素都在同一个名称空间(“http://www.address.org”)中。

相反，假设Person类的属性是Address对象。

使用NAMESPACE参数将Person类分配给名称空间“http://www.person.org”。

如果导出Person类的一个对象，将收到如下输出：

```
<Person xmlns="http://www.person.org">
  <Name>Zevon, Samantha H.</Name>
  <DOB>1964-05-24</DOB>
  <Address xmlns="http://www.address.org">
    <Street>8280 Main Avenue</Street>
    <City>Washington</City>
    <State>VT</State>
    <Zip>15355</Zip>
  </Address>
</Person>
```

注意，<Address>元素位于其父对象(“http://www.person.org”)的名称空间中。

但是，<Address>的元素位于名称空间“http://www.address.org”中。

控制局部元素是否限定

在顶层导出对象时，通常将其视为全局元素。然后根据启用XML的对象的ELEMENTQUALIFIED参数的设置处理其本地元素。如果未设置此类参数，则改用编写器属性ElementQualified的值；默认情况下，文本格式为1，编码格式为0。

下面的示例显示一个默认设置为ElementQualified的对象，即1：

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<PersonNS xmlns="http://www.person.com" GroupID="M9301">
  <Name>Pybus, Gertrude X.</Name>
  <DOB>1986-10-19</DOB>
</PersonNS>
```

该名称空间被添加到<PersonNS>元素中作为默认名称空间，因此应用于元素<Name>和<DOB>子元素。

我们修改了写入器定义并将ElementQualified属性设置为0。
在本例中，相同的对象如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<s01:PersonNS xmlns:s01="http://www.person.com" GroupID="M9301">
  <Name>Pybus, Gertrude X.</Name>
  <DOB>1986-10-19</DOB>
</s01:PersonNS>
```

在本例中，名称空间被添加到带有前缀的<PersonNS>元素中，该前缀用于<PersonNS>元素，但不用于其子元素。

控制一个元素是否局部于它的父元素

默认情况下，当使用object()方法生成一个元素并且该元素具有命名空间时，该元素不是其父元素的本地元素。相反，可以强制元素属于其父元素的命名空间。为此，可以使用object()方法的可选本地参数；这是第四个参数。

本地参数为0(默认值)

在这里的示例中虑将NAMESPACE类参数指定为 “ http://www.person.com ” 的Person类。

如果打开根元素，然后使用Object()生成Person，则<Person>元素位于 “ http://www.person.com ” 名称空间中。
以下例子：

```
<?xml version="1.0" encoding="UTF-8"?>
<Root xmlns="http://www.rootns.org">
  <Person xmlns="http://www.person.com">
    <Name>Adam, George L.</Name>
    <DOB>1947-06-29</DOB>
  </Person>
</Root>
```

如果我们将<Person>元素更深地嵌套到其他元素中，也会出现类似的结果。

```
<?xml version="1.0" encoding="UTF-8"?>
<Root xmlns="www.rootns.org">
  <GlobalEl xmlns="globalns">
    <Inner xmlns="innerns">
      <Person xmlns="http://www.person.com">
        <Name>Adam, George L.</Name>
        <DOB>1947-06-29</DOB>
      </Person>
    </Inner>
  </GlobalEl>
</Root>
```

局部参数设置为1

为了强制<Person>元素成为其父元素的本地元素，我们将local参数设置为1。
如果我们这样做并再次生成前面的输出，我们将收到以下嵌套版本的输出：

```
<?xml version="1.0" encoding="UTF-8"?>
<Root xmlns="http://www.rootns.org">
  <s01:Person xmlns="http://www.person.com"
xmlns:s01="http://www.rootns.org">
    <Name>Adam, George L.</Name>
    <DOB>1947-06-29</DOB>
  </s01:Person>
</Root>
```

注意，现在<Person>元素在“http://www.rootns.org”名称空间中，这是它的父元素的名称空间。
类似地，嵌套程度更高的版本应该是这样的：

```
<?xml version="1.0" encoding="UTF-8"?>
<Root xmlns="www.rootns.org">
  <GlobalEl xmlns="globalns">
    <Inner xmlns="innerns">
      <s01:Person xmlns="http://www.person.com" xmlns:s01="innerns">
        <Name>Adam, George L.</Name>
        <DOB>1947-06-29</DOB>
      </s01:Person>
    </Inner>
  </GlobalEl>
</Root>
```

控制属性是否限定

导出对象时，默认情况下其属性不合格。要使它合格，请将编写器属性AttributeQualified设置为1。下面的示例显示AttributeQualified等于0(或尚未设置)的编写器生成的输出：

```
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <s01:Person xmlns:s01="http://www.person.com" GroupID="E8401">
    <Name>Leiberman, Amanda E.</Name>
    <DOB>1988-10-28</DOB>
  </s01:Person>
</Root>
```

相反，下面的示例显示使用AttributeQualified等于1的编写器生成的同一对象：

```
<?xml version="1.0" encoding="UTF-8"?>
<Root>
  <s01:Person xmlns:s01="http://www.person.com" s01:GroupID="E8401">
    <Name>Leiberman, Amanda E.</Name>
    <DOB>1988-10-28</DOB>
  </s01:Person>
```

</Root>

在这两种情况下，元素都是不合格的。

命名空间分配摘要

本节介绍如何为XML输出中的任何给定元素确定命名空间。

顶级元素

对于与在顶级导出的InterSystems IRIS类相对应的元素，适用以下规则：

1. 如果为类指定了Namespace参数，则元素位于该命名空间中。
2. 如果未指定该参数，则元素位于在生成元素的输出方法(RootObject()、RootElement()、Object()或Element())中指定的命名空间中。
3. 如果未在输出方法中指定命名空间，则元素位于编写器的DefaultNamespace属性指定的命名空间中。
4. 如果DefaultNamespace属性为空，则元素不在任何命名空间中。

低层元素

要导出的类的子元素受该类的ELEMENTQUALIFIED参数影响。如果未设置ELEMENTQUALIFIED，则改用编写器属性ElementQualified的值；默认情况下，文本格式为1，编码格式为0。

如果元素符合给定类的条件，则该类的子元素将按如下方式分配给命名空间：

1. 如果为父对象指定了Namespace参数，则子元素将显式分配给该命名空间。
2. 如果未指定该参数，子元素将显式分配给在生成元素的输出方法(RootObject()、RootElement()、Object()或Element())中指定的命名空间。
3. 如果未在输出方法中指定命名空间，则子元素将显式分配给由编写器的DefaultNamespace属性指定的命名空间。
4. 如果DefaultNamespace属性为空，则子元素不会显式分配给任何命名空间。

[#Caché #InterSystems IRIS](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AD%E7%AB%A0-%E6%8E%A7%E5%88%B6%E5%90%8D%E7%A7%B0%E7%A9%BA%E9%97%B4%E7%9A%84%E4%BD%BF%E7%94%A8>