

ValidateThis! - Less Verbose XML

Posted At : October 19, 2008 5:16 PM | Posted By : Bob Silverberg

Related Categories: ColdFusion, ValidateThis

I was creating an **XML schema definition** for my current XML schema and I realized that I could make it much less verbose. I could eliminate most of the "container" elements, such as *rules* and *params* and it wouldn't affect its readability or parseability, and would save me some typing when creating the XML files, so I've done that.

For the record, here is the new sample XML file:

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

<validateThis xsi:noNamespaceSchemaLocation="validateThis.xsd"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <objectProperties>

    <property name="UserName" desc="Email Address">

      <rule type="required" contexts="Register,Update" />

      <rule type="email" contexts="Register,Update"

        failureMessage="Hey, buddy, you call that an Email Address?" />

    </property>

    <property name="Nickname">

      <rule type="custom" contexts="Register,Update">

        <param methodName="CheckDupNickname" />

      </rule>

    </property>

    <property name="UserPass" desc="Password">

      <rule type="required" contexts="Register,Update" />

      <rule type="rangelength" contexts="Register,Update">

        <param minLength="5" />

        <param maxLength="10" />

      </rule>

      <rule type="equalTo" contexts="Register,Update">

        <param ComparePropertyName="VerifyPassword" />

      </rule>

    </property>

    <property name="VerifyPassword" desc="Verify Password">

      <rule type="required" contexts="Register,Update" />

    </property>

    <property name="FirstName" desc="First Name">

      <rule type="required" contexts="Update" />

    </property>

    <property name="LastName" desc="Last Name">

      <rule type="required" contexts="Update" />

      <rule type="required" contexts="Register">

        <param DependentPropertyName="FirstName" />

      </rule>

    </property>

    <property name="LikeOther" desc="What do you like?">

      <rule type="required" contexts="Register,Update"

        failureMessage="If you don't like Cheese and you don't like Chocolate, you must like something!">

        <param ServerCondition="getLikeCheese() EQ 0 AND getLikeChocolate() EQ 0" />

        <param ClientCondition="&quot;[name='LikeCheese']&quot;.getValue() == 0 &amp;&amp; &quot;[name='LikeChocolate']&quot;.getValue() == 0;" />

      </rule>

    </property>

    <property name="HowMuch" desc="How much money would you like?">

      <rule type="numeric" contexts="Register,Update" />

    </property>

    <property name="AllowCommunication" desc="Allow Communication" />

    <property name="CommunicationMethod" desc="Communication Method">

      <rule type="required" contexts="Register,Update"

        failureMessage="If you are allowing communication, you must choose a communication method.">

        <param DependentPropertyName="AllowCommunication" />

        <param DependentPropertyValue="1" />

      </rule>

    </property>

  </objectProperties>

</validateThis>
```

```

    </rule>

  </property>

</objectProperties>

</validateThis>

```

And here's the xsd, without the annotations that I've since added to it:

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="validateThis">

    <xs:complexType>

      <xs:sequence>

        <xs:element name="objectProperties">

          <xs:complexType>

            <xs:sequence>

              <xs:element

                maxOccurs="unbounded"

                name="property">

                  <xs:complexType>

                    <xs:sequence>

                      <xs:element

                        minOccurs="0"

                        maxOccurs="unbounded"

                        name="rule">

                          <xs:complexType>

                            <xs:sequence>

                              <xs:element

                                minOccurs="0"

                                maxOccurs="unbounded"

                                name="param">

                                  <xs:complexType>

                                    <xs:anyAttribute

                                      namespace="##any"

                                      processContents="lax"/>

                                  </xs:complexType>

                                </xs:element>

                              </xs:sequence>

                              <xs:attribute

                                name="type"

                                type="xs:string"

                                use="required"/>

                              <xs:attribute

                                name="contexts"

                                type="xs:string"

                                use="optional"/>

                              <xs:attribute

                                name="failureMessage"

                                type="xs:string"

                                use="optional"/>

                            </xs:complexType>

                          </xs:element>

                        </xs:sequence>

                      <xs:attribute

                        name="name"

                        type="xs:string"

                        use="required"/>

                      <xs:attribute

                        name="desc"

                        type="xs:string"

                        use="optional"/>

```

```
</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:sequence>

</xs:complexType>

</xs:element>

</xs:schema>
```