

Suche in -> Deutsche Artikel

Welcher VERFAHRENSWEISE: TO: Sie überprüfen XML-Fragmente anhand eines XML-Schemas in visuellem C# .NET

Wichtig: Dieser Artikel wurde maschinell und nicht von einem Menschen übersetzt. Die Microsoft Knowledge Base ist sehr umfangreich und ihre Inhalte werden ständig ergänzt beziehungsweise überarbeitet. Um Ihnen dennoch alle Inhalte auf Deutsch anbieten zu können, werden viele Artikel nicht von Menschen, sondern von Übersetzungsprogrammen übersetzt, die kontinuierlich optimiert werden. Doch noch sind maschinell übersetzte Texte in der Regel nicht perfekt, insbesondere hinsichtlich Grammatik und des Einsatzes von Fremdwörtern sowie Fachbegriffen. Microsoft übernimmt keine Gewähr für die sprachliche Qualität oder die technische Richtigkeit der Übersetzungen und ist nicht für Probleme haftbar, die direkt oder indirekt durch Übersetzungsfehler oder die Verwendung der übersetzten Inhalte durch Kunden entstehen könnten.

Auf dieser Seite

- ↓ [Zusammenfassung](#)
- ↓ [Voraussetzungen](#)
- ↓ [Erstellen Sie ein XSD](#)
- ↓ [Erstellen Sie visuelle C#-Anwendung in .NET](#)
- ↓ [Informationsquellen](#)

Artikel-ID : 318504
Geändert am : Dienstag, 16. September 2003
Version : 3.1

Zusammenfassung

Dieser Artikel beschreibt Schritt für Schritt, wie `XmlValidatingReader`- und `XMLSchemaCollection`-Objekte zu Überprüfung eines XML-Fragments für ein XML-Schema verwandt werden.

XmlValidatingReader implementiert die **XmlReader** Klasse und bietet Unterstützung für XML-Datenüberprüfung. Die **Schemas** Eigenschaft von **XmlValidatingReader** verbindet den Leser in den Schemadateien, die in einem **XmlSchemaCollection** zwischengespeichert werden. Die **ValidationType** Eigenschaft von **XmlValidatingReader** gibt den Typen der Überprüfung, die der Leser ausführen sollte, an. Wenn Sie die Eigenschaft zu **ValidationType.None** festlegen, erstellen Sie einen nonvalidating Leser.

Nur können Sie `XmlSchemaCollection` XML Schema Definition Language (XSD) Schemas und XDR-XDR-Schemas hinzufügen. Verwenden Sie zu Laden von Schemas die `Add` Methode mit einem Namespace-Uri. Das typische Namespace-Uri ist `TargetNamespace` die Eigenschaft des Schemas für XML-Schemas.

Voraussetzungen

In der folgenden Liste sind die empfohlene Hardware, Software, Netzwerkinfrastruktur und erforderlichen Service Packs aufgeführt

- Microsoft Visual Studio .NET das auf kompatiblen Microsoft Windows-Betriebssystemen installiert wird

In diesem Artikel wird vorausgesetzt, dass Sie über Erfahrungen auf den folgenden Gebieten verfügen

- Visuelles C#-.NET
- Grundlegende XML-Standards
- XSD-Schemas

Erstellen Sie ein XSD

Fügen Sie den folgenden Code in einer neuen Textdatei mit Name `C:\Books.xsd` ein:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="urn:bookstore-schema"
  elementFormDefault="qualified" targetNamespace="urn:bookstore-schema">
  <xsd:element name="bookstore" type="bookstoreType" />
  <xsd:element name="comment" type="xsd:string" />
  <xsd:element name="author" type="authorName"/>
  <xsd:complexType name="authorName">
    <xsd:sequence>
      <xsd:element name="first-name" type="xsd:string" />
      <xsd:element name="last-name" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="bookstoreType">
    <xsd:sequence maxOccurs="unbounded">
      <xsd:element name="book" type="bookType" />
      <xsd:element ref="comment" minOccurs="0" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="bookType">
    <xsd:sequence>
      <xsd:element name="title" type="xsd:string" />
      <xsd:element ref="author" />
      <xsd:element name="price" type="xsd:decimal" />
      <xsd:attribute name="genre" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Erstellen Sie visuelle C#-Anwendung in .NET

1. Erstellen Sie eine neue Anwendung von Visual C# .NET Console.
2. Ersetzen Sie den Code in "Class1.cs" durch folgenden Code

```
using System; using System.Xml; using System.Xml.Schema; namespace ConsoleApplication3 {  
    /// <summary> /// Summary description for Class1. /// </summary> class Class1 { ///  
    <summary> /// The main entry point for the application. /// </summary> System.Boolean  
    m_success; [STAThread] static void Main(string[] args) { // TODO: Add code to start  
    application here. // XmlValidatingReader reader = null; XmlSchemaCollection myschema =  
    new XmlSchemaCollection(); ValidationEventHandler eventHandler = new  
    ValidationEventHandler(Class1.ShowCompileErrors); try { //Create the XML fragment to be  
    parsed. String xmlFrag = "<author xmlns='urn:bookstore-schema'  
    xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>" + "<first-name>Herman</first-  
    name>" + "<last-name>Melville</last-name>" + "</author>"; //Create the XmlParserContext.  
    XmlParserContext context = new XmlParserContext(null, null, "", XmlSpace.None);  
    //Implement the reader. reader = new XmlValidatingReader(xmlFrag, XmlNodeType.Element,  
    context); //Add the schema. myschema.Add("urn:bookstore-schema", "c:\\Books.xsd"); //Set  
    the schema type and add the schema to the reader. reader.ValidationType =  
    ValidationType.Schema; reader.Schemas.Add(myschema); while (reader.Read()) { }  
    Console.WriteLine("Completed validating xmlfragment"); } catch (XmlException XmlExp) {  
    Console.WriteLine(XmlExp.Message); } catch (XmlSchemaException XmlSchExp) {  
    Console.WriteLine(XmlSchExp.Message); } catch (Exception GenExp) {  
    Console.WriteLine(GenExp.Message); } finally { Console.Read(); } } public static void  
    ShowCompileErrors(object sender, ValidationEventArgs args) {  
    Console.WriteLine("Validation Error: {0}", args.Message); } } }
```

3. Das XML-Fragment ist ein gültiges Element, wenn die folgende Meldung in dem Ausgabenfenster angezeigt wird:

```
Completed validating xmlfragment
```

Hinweis: nur die Typendeklarationen und die Elemente der oberste Ebene in dem XML-Schema werden für ein XML-Schema validiert. Teilelement werden als lokal betrachtet und können deshalb nicht bestätigt werden. Deklarieren Sie ein Element der oberste Ebene und verweisen Sie dazu, um ein Teilelement zu überprüfen.

Informationsquellen

Erhalten Sie Weitere Informationen, wenn unten auf Artikelnummer klickt, um die Beiträge in dem Microsoft Knowledge Base anzuzeigen:

[307379](http://support.microsoft.com/kb/307379/EN-US/) (http://support.microsoft.com/kb/307379/EN-US/) Welcher VERFAHRENSWEISE: TO: indem Sie bei visuellem C#-.NET DTD, XDR oder XSD verwenden, überprüfen Sie XML-Dokument

[313826](http://support.microsoft.com/kb/313826/EN-US/) (http://support.microsoft.com/kb/313826/EN-US/) INFO: Wegweiser für XML-Schema in dem .NET

[313651](http://support.microsoft.com/kb/313651/EN-US/) (http://support.microsoft.com/kb/313651/EN-US/) INFO: Wegweiser für XML in dem .NET

Weitere Informationen finden Sie auf den folgenden MSDN-Websites:

System.Xml-Namespaces

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfsystemxml.asp>
(http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfsystemxml.asp)

XML-Webdienst

<http://msdn.microsoft.com/xml> (http://msdn.microsoft.com/xml)

Die Informationen in diesem Artikel beziehen sich auf:

- Microsoft Visual C# .NET 2003 Standard Edition
- Microsoft Visual C# .NET 2002 Standard Edition
- Microsoft .NET Framework 1.0
- Microsoft .NET Framework 1.1

Keywords: kbhowtomaster KB318504 KbMtde kbmt

Microsoft stellt Ihnen die in der Knowledge Base angebotenen Artikel und Informationen als Service-Leistung zur Verfügung. Microsoft übernimmt keinerlei Gewährleistung dafür, dass die angebotenen Artikel und Informationen auch in Ihrer Einsatzumgebung die erwünschten Ergebnisse erzielen. Die Entscheidung darüber, ob und in welcher Form Sie die angebotenen Artikel und Informationen nutzen, liegt daher allein bei Ihnen. Mit Ausnahme der gesetzlichen Haftung für Vorsatz ist jede Haftung von Microsoft im Zusammenhang mit Ihrer Nutzung dieser Artikel oder Informationen ausgeschlossen.