

27.3 Klasse XmlReader

Die Klasse XmlReader (gb.xml) repräsentiert einen XML-Stream-Reader, der die SAX-API implementiert. Diese Klasse beerbt die virtuelle Klasse `_XmlReader`.

Ein Objekt der Klasse XmlReader (gb.xml) können Sie erzeugen:

```
Dim hXmlReader As XmlReader
hXmlReader = New XmlReader()
```

27.3.1 Eigenschaften

Wesentliche Eigenschaften der Klasse `XmlReader`:

Eigenschaft	Datentyp	Beschreibung
Depth	Integer	Gibt die Such-Tiefe in Bezug auf den aktuell gelesenen Knoten - relativ zum XML-Baum - an.
Eof	Boolean	Gibt True zurück, wenn mit <code>XmlReader.Read()</code> das Ende der XML-Datei erreicht wurde.
Node	<code>.XmlReader.Node</code>	Gibt Informationen über den aktuell gelesenen Knoten zurück. Genutzt werden dazu die Eigenschaften der virtuellen Klasse <code>.XmlReader.Node</code> .
Pos	Integer	Gibt die Position im XML-Stream zurück, an dem der interne Zeiger steht.
ReadFlags	<code>.XmlReaderReadFlags</code>	Gibt eine virtuelle Klasse zurück, die sich wie ein Schreib-/Lese-Array verwenden lässt.
StoredNodes	<code>XmlNode[]</code>	Gibt ein Array mit allen Knoten im XML-Stream zurück.
InputStream	Stream	<code>XmlReader.InputStream.Lines</code> gibt ein virtuelles Objekt zurück, das es Ihnen ermöglicht durch einen Stream Zeile für Zeile zu iterieren.

Tabelle 27.3.1.1 : Eigenschaften der Klasse XmlReader

Im folgenden Beispiel sehen Sie, wie Sie auf einfache Art den Inhalt einer XML-Datei in einer TextArea anzeigen können:

```
Public xmlReader As XmlReader
Public sXMLPath As String = "files/example.xml"

Public Sub Form_Open()

    Dim sLine As String

    xmlReader = New XmlReader
    xmlReader.Open(sXMLPath)
    For Each sLine In XmlReader.InputStream.Lines
        txaXML.Insert(sLine & gb.NewLine)
    Next
    xmlReader.Close()

    txaXML.Pos = 0

End
```

Wenn Sie sich beim sequentiellen Lesen einer XML-Datei nur für die Attribute interessieren, dann können Sie ausgewählte Knoten-Typen wie Elemente, EndElement und Text so ausblenden:

```
XmlReader.ReadFlags[XmlReaderNodeType.Attribute] = True
XmlReader.ReadFlags[XmlReaderNodeType.Element] = False
XmlReader.ReadFlags[XmlReaderNodeType.Text] = False
XmlReader.ReadFlags[XmlReaderNodeType.EndElement] = False
```

Unter <https://gambas-playground.proko.eu/?gist=b90eeff3dacbec0548d01701f3c05133> hat Adrien Prokopowicz ein passendes interaktives Beispiel hinterlegt.

27.3.1.1 Eigenschaften der virtuelle Klasse `.XmlReader.Node`

Hier finden Sie die Übersicht zu den Eigenschaften der virtuellen Klasse `.XmlReader.Node`

Eigenschaft	Datentyp	Beschreibung
IsEmptyElement	Boolean	Gibt True zurück, wenn das Element leer ist.
Name	String	Gibt den Namen des aktuellen Attribut-Knotens zurück.
Type	Integer	Gibt den Typ des aktuellen Attribut-Knotens zurück.
Value	String	Gibt den Wert des aktuellen Attribut-Knotens zurück.
Attributes	.XmlReader.No- de.Attributes	Gibt eine virtuelle Collection aller Attribut-Knoten des aktuellen Knotens zurück.

Tabelle 27.3.1.1.1 : Eigenschaften der Klasse XmlReader.Node

Die Klasse XmlReader.Node verhält sich wie ein Nur-Lesen-Array:

```
Dim hXmlReader.Node As XmlReader.Node
Dim sString As String
sString = hXmlReader.Node.Attributes[Name As String]
```

und ist aufzählbar über eine For-Each-Kontrollstruktur:

```
Dim hXmlReader.Node As XmlReader.Node
Dim sString As String
For Each sString In hXmlReader.Node.Attributes
    ...
Next
```

27.3.1.2 Eigenschaften der virtuelle Klasse .XmlReader.Node.Attributes

Die virtuelle Klasse .XmlReader.Node.Attributes verfügt über drei Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Count	Integer	Gibt die Anzahl der Attribute für den aktuellen Knoten zurück.
Name	Integer	Gibt den Namen des aktuellen Attributs zurück. Die Zahl repräsentiert eine Attribut-Name-Konstante → Klasse XmlReaderNodeType
Value	Variant	Gibt den Wert des aktuellen Attributs zurück.

Tabelle 27.3.1.2.1 : Eigenschaften der Klasse XmlReader.Node.Attributes

Die Klasse XmlReader.Node.Attributes besitzt nur eine Methode:

```
Function Exist(Name As String) As Boolean
```

Die Funktion gibt True zurück, wenn ein Attribut mit dem angegebenen Namen 'Name' für den aktuellen Knoten existiert.

Die Klasse XmlReaderNodeType ist statisch und enthält nur Konstanten:

```
Attribute (8), CDATA (4), Comment (3), Document (0), DocumentFragment (0), DocumentType (0), Element (0), EndElement (6), EndEntity (0), EndStream (7), Entity (0), EntityReference (0), None (0), Notation (0), ProcessingInstruction (0), SignificantWhitespace (0), Text (2), Whitespace (0), XmlDeclaration (0)
```

27.3.2 Methoden

Hier finden Sie die Übersicht zu den vier Methoden der Klasse *XmlReader*:

Methode	Rückgabotyp	Beschreibung
Open(path As String)	-	Öffnet eine XML-Datei mit dem angegebenen Datei-Pfad 'path'.
Close()	-	Schließt eine Datei, die mit der Open()-Methode geöffnet wurde.
FromString(Content As String)	-	Überschreibt den Inhalt eines XmlReader-Objektes mit dem Inhalt von 'Content'.
Read()	Integer	Liest einen Knoten und gibt ihn zurück.

Tabelle 27.3.2.1 : Ausgewählte Methoden der Klasse XmlReader

27.3.3 Beispiel XML-Parser (XmlReader)

Im vorgestellten Beispiel sollen aus Kontaktdaten – gespeichert in einer XML-Datei – ausgewählte Daten ausgelesen werden und dann in einer TextArea angezeigt werden. Die ausgelesenen, aufbereiteten und in einer Text-Datei abgespeicherten Daten könnten zum Beispiel als Basis für den Druck von Adress-Aufklebern dienen:

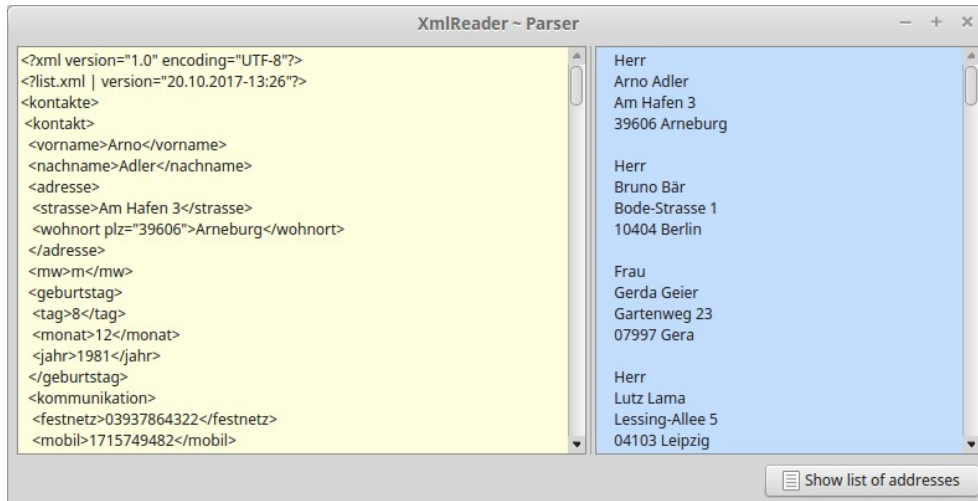


Abbildung 27.3.3.1: GUI des Beispiel-Projektes 'XML-Parser'

Der Quelltext wird vollständig angegeben, wobei Sie besonders auf den Quelltext der Funktion `GetRecords()` achten sollten:

```
' Gambas class file

Public hXMLReader As XmlReader
Public sXMLPath As String = "files/list.xml"

Public Sub Form_Open()
  ShowXMLContent()
  txaXML.Pos = 0
  HSplit1.Layout = [3, 2]
End

Public Sub btnShowRecords_Click()

  Dim i, j As Integer
  Dim asMatrix As String[]
  Dim avRecords As New Variant[]
  Dim sSpace As String = String$(4, " ")

  avRecords = GetRecords()

  For i = 0 To avRecords.Max
    asMatrix = New String[]
    asMatrix = avRecords[i]
    txaList.Insert(sSpace & asMatrix[0] & gb.NewLine)
    txaList.Insert(sSpace & asMatrix[1] & " " & asMatrix[2] & gb.NewLine)
    txaList.Insert(sSpace & asMatrix[3] & gb.NewLine)
    txaList.Insert(sSpace & asMatrix[4] & " " & asMatrix[5] & gb.NewLine)
    txaList.Insert(gb.NewLine)
  Next

  File.Save(Application.Path & / "files/addresslist.txt", txaList.Text)
End

Private Sub ShowXMLContent()

  Dim sLine As String

  hXMLReader = New XmlReader
  hXMLReader.Open(sXMLPath)
  For Each sLine In hXMLReader.InputStream.Lines
    txaXML.Insert(sLine & gb.NewLine)
  Next
  hXMLReader.Close()
```

```

End

Private Function GetRecords() As Variant[]

    Dim asRecord As String[]
    Dim avRecords As New Variant[]

    txaList.Clear()
    asRecord = New String[]

    hXMLReader = New XmlReader
    hXMLReader.Open(sXMLPath)
    hXMLReader.Read()

    While Not hXMLReader.Eof
        Select Case hXMLReader.Node.Type
            Case XMLReaderNodeType.Element
                If hXMLReader.Node.Name = "mw" Then
                    hXMLReader.Read()
                    If hXMLReader.Node.Value = "w" Then
                        asRecord.Add("Frau", 0)
                    Else
                        asRecord.Add("Herr", 0)
                    Endif
                    avRecords.Add(asRecord)
                    asRecord = New String[]
                Endif
                If hXMLReader.Node.Name = "vorname" Then
                    hXMLReader.Read()
                    asRecord.Add(hXMLReader.Node.Value, 1)
                Endif
                If hXMLReader.Node.Name = "nachname" Then
                    hXMLReader.Read()
                    asRecord.Add(hXMLReader.Node.Value, 2)
                Endif
                If hXMLReader.Node.Name = "strasse" Then
                    hXMLReader.Read()
                    asRecord.Add(hXMLReader.Node.Value, 3)
                Endif
                If hXMLReader.Node.Name = "wohnort" Then
                    For Each hXMLReader.Node.Attributes ' Zuerst die Attribute auslesen!
                        Select Case hXMLReader.Node.Attributes.Name
                            Case "plz"
                                asRecord.Add(hXMLReader.Node.Attributes.Value, 4)
                        End Select
                    Next
                    hXMLReader.Read()
                    asRecord.Add(hXMLReader.Node.Value, 5)
                Endif
            End Select
        hXMLReader.Read()
    Wend
    hXMLReader.Close()

    Return avRecords

End

```