

24.12.1 RSS-Feed-Projekte

Die Komponente `gb.web.feed` – kurz `WebFeed` – entspricht der RSS-2.0-Spezifikation und bildet die RSS-Spezifikation 1:1 ab. Das bedeutet, dass

- nur RSS-Feeds, die der Spezifikation entsprechen, fehlerfrei geparkt werden und
- valide RSS-Feeds nur dann erzeugt werden, wenn die Spezifikation beachtet wird!

24.12.1.1 Projekt 1

Das erste Projekt erzeugt einen leeren RSS-Feed:

```
Public hRSS As Rss

Private Function WriteFeed() As String

    Dim hXML As String
    Dim hRSSItem As RssItem

    '-- Es wird ein neues RSS-Objekt erzeugt
    hRSS = New Rss

    '-- Der Feed wird gerendert → XML-Format
    hXML = hRSS.ToString()

    Return hXML

End
```

Das ist der Inhalt von `hXML` – wie Sie erkennen, ist es ein valides RSS-Dokument:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
  <channel>
    <title />
    <link />
    <description />
  </channel>
</rss>
```

24.12.1.2 Projekt 2

Im zweiten Projekt werden in das o.a. Template in den Kanal zwei Einträge eingefügt. Dann wird das XML-Dokument in der Datei `rss.xml` (temporär) abgespeichert. Anschließend kann die Datei geladen, der Inhalt eingelesen und geparkt sowie ausgewählte Daten ausgegeben werden:

```
' Gambas class file

Public hRSS As Rss

Public Sub Form_Open()
    FMain.Caption = "Schreiben und Lesen eines RSS-Feeds → Datei ~/rss.xml"
End

Public Sub btnWriteFeed_Click()

    WriteFeed()
    txaOutput.Insert(File.Load(Temp("rss.xml")))

End

Public Sub btnReadFeed_Click() '
    '-- Der RSS-Feed wird aus einer XML-Datei geladen - wenn sie existiert
    If Not Exist(Temp("rss.xml")) Then Return

    ReadFeed(File.Load(Temp("rss.xml")))

    txaOutput.SetFocus()
    txaOutput.Pos = txaOutput.Length

End

Public Sub btnClear_Click()
    txaOutput.Clear()
End
```

```

Private Function WriteFeed() As String
    Dim hXML As String
    Dim hRSSItem As RssItem

    '-- Es wird ein neues RSS-Objekt erzeugt
    hRSS = New Rss

    '-- Ausgewählte Channel-Eigenschaften werden festgelegt
    hRSS.Title = "CHANNEL 1 - TITEL"
    hRSS.Link = "CHANNEL 1 - LINK: www.channel.net"
    hRSS.Description = "CHANNEL 1 - DESCRIPTION"
    hRSS.Pub = New RssDate(DateAdd(Now, gb.Hour, 0))

    '-- Es wird ein neuer Eintrag 1 erzeugt
    hRSSItem = New RssItem
    '-- Ausgewählte Item-Eigenschaften werden festgelegt
    hRSSItem.Title = "ITEM 1 - TITEL"
    hRSSItem.Link = "ITEM 1 - LINK 1: www.example.com/item_1"
    hRSSItem.Description = "ITEM 1 - DESCRIPTION"
    hRSSItem.Pub = New RssDate
    '-- Der Eintrag 1 wird in den Feed eingefügt
    hRSS.Add(hRSSItem)

    '-- Es wird ein neuer Eintrag 2 erzeugt
    hRSSItem = New RssItem
    '-- Ausgewählte Item-Eigenschaften werden festgelegt
    hRSSItem.Title = "ITEM 2 - TITEL"
    hRSSItem.Link = "ITEM 2 - LINK 2: www.example.com/item_2"
    hRSSItem.Description = "ITEM 2 - DESCRIPTION"
    hRSSItem.Pub = New RssDate
    '-- Der Eintrag 2 wird in den Feed eingefügt
    hRSS.Add(hRSSItem)

    '-- Der Feed wird gerendert -> XML-Format
    hXML = hRSS.ToString()

    '-- Der RSS-Feed wird in einer (temporären) XML-Datei gespeichert
    File.Save(Temp("rss.xml"), hXML)

    Return hXML
End

Private Function ReadFeed(argData As String)

    Dim hXML As String
    Dim iRSSCount As Integer
    Dim hRSSItem As RssItem

    '-- Der RSS-Feed wird aus einer XML-Datei geladen - wenn sie existiert
    If Not Exist(Temp("rss.xml")) Then Return

    hXML = File.Load(Temp("rss.xml"))

    '-- Es wird ein neues RSS-Objekt erzeugt
    hRSS = New Rss
    hRSS.FromString(hXML)

    '-- Ausgewählte Channel-Eigenschaften werden ausgelesen *und* angezeigt
    If hRSS.Title Then txaOutput.Insert(hRSS.Title & gb.NewLine)
    If hRSS.Link Then txaOutput.Insert(hRSS.Link & gb.NewLine)
    If hRSS.Description Then txaOutput.Insert(hRSS.Description & gb.NewLine)
    If hRSS.Pub Then txaOutput.Insert(Format(hRSS.Pub.Date, "dddd, dd.mm.yyyy hh:nn") & gb.Lf & gb.Lf)

    '-- Ausgewählte Item-Eigenschaften werden ausgelesen *und* angezeigt
    For Each hRSSItem In hRSS
        If hRSSItem.Title Then txaOutput.Insert(hRSSItem.Title & gb.NewLine)
        If hRSSItem.Link Then txaOutput.Insert(hRSSItem.Link & gb.NewLine)
        If hRSSItem.Description Then txaOutput.Insert(hRSSItem.Description & gb.NewLine)
        Inc iRSSCount
        If hRSSItem.Pub Then
            If iRSSCount < hRSS.Count Then
                txaOutput.Insert(Format(hRSSItem.Pub.Date, "dddd, dd.mm.yyyy hh:nn") & gb.NewLine & gb.NewLine)
            Else
                txaOutput.Insert(Format(hRSSItem.Pub.Date, "dddd, dd.mm.yyyy hh:nn") & gb.NewLine)
            Endif
        Endif
    Next
End

```

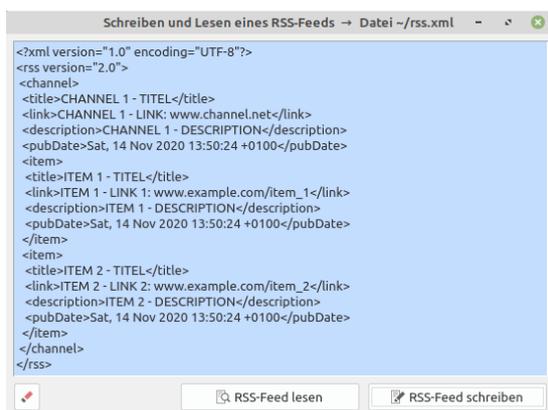


Abbildung 24.12.1.1: Inhalt RSS-Feed

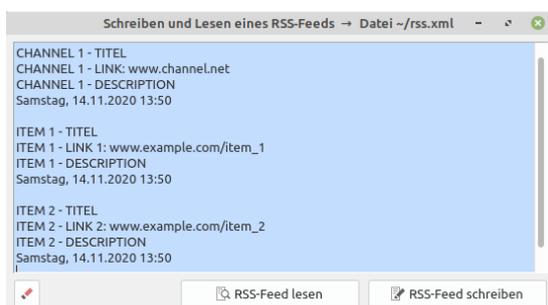


Abbildung 24.12.1.2: Anzeige ausgewählter Daten im RSS-Feed

Den Quelltext des Projektes 2 finden Sie im Download-Bereich.

24.12.1.3 Projekt 3

Das Projekt 3 basiert auf dem Projekt 2. Die Erweiterung besteht darin, dass ausgewählte Daten in einer ColumnView angezeigt werden. Die bisher für solche Einsatzfälle benutzte Komponente ListContainer gilt als veraltet und sollte für aktuelle Projekte nicht mehr genutzt werden. Nach einem Klick auf eine Zeile in der ColumnView wird in einer WebView die Webseite angezeigt, deren URL in der Eigenschaft *RSSItem.Link* hinterlegt ist.

Hinweise:

- Der ausgewählte RSS-Feed enthält proprietäre XML-Tags in eigenen Namensräumen, die nicht der RSS-2.0-Spezifikation entsprechen. Dadurch werden beim Parsen Fehler erzeugt, wenn die Komponente *gb.web.feed* benutzt wird.
- Um Ihnen das Projekt 3 mit der besonderen Art der Daten-Präsentation in einer ColumnView und in einer WebView zu zeigen, wurde ein originaler RSS-Feed gekürzt, alle proprietären XML-Tags entfernt und der Feed in einer XML-Datei gespeichert. Das Parsen gelingt somit fehlerfrei, weil das Dokument nun der Spezifikation entspricht.
- Die Komponente *gb.web.feed* wird gegenwärtig überarbeitet. An der Umsetzung der RSS-2.0-Spezifikation wird festgehalten. Es ist aber vorgesehen, dass proprietäre XML-Tags zukünftig ignoriert werden und dann keine Fehler mehr beim Parsen erzeugen.

Hinweis:

Das Projekt in https://gambas-buch.de/doku.php?id=k18:k18.12:start#projekt_rss-reader kann als Alternative für den Fall angesehen werden, dass auch proprietäre XML-Tags – die bekannt sein müssen – verarbeitet werden sollen.

Auch den Quelltext des Projektes 3 finden Sie im Download-Bereich.