

23.9.6 MediaMessage (gb.media)

Die Klasse MediaMessage kann nicht erzeugt werden. Die Klasse verhält sich wie ein Nur-Lesen-Array und ist mit dem Schlüsselwortpaar 'For Each' aufzählbar. So wird der Wert zurück gegeben, der mit einem in der Nachricht gespeicherten Schlüssel-Namen verknüpft ist:

```
Dim hMediaMessage As MediaMessage
Dim vValue As Variant

vValue = hMediaMessage [ Name As String ]
```

Alle in einer Nachricht gespeicherten Werte werden so aufgezählt:

```
Dim hMediaMessage As MediaMessage
Dim vValue As Variant

For Each vValue In hMediaMessage
    ...
Next
```

23.9.6.1 Eigenschaften

Die Klasse MediaMessage verfügt über diese fünf Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Count	Integer	Gibt die Anzahl der in der Nachricht gespeicherten Schlüssel-/Wertepaare zurück.
Name	String	Gibt den Namen der Nachricht zurück. Er kann auch NULL sein!
Keys	String[]	Gibt ein Array mit den <u>Schlüsseln</u> aller in der Nachricht gespeicherten Werte zurück.
Key	String	Gibt den Schlüssel des letzten in der Nachricht gespeicherten Wertes zurück.
Type	Integer	Gibt den Typ des Ereignisses als MediaMessage-Konstante zurück.

Tabelle 23.9.6.1.1 : Eigenschaften der Klasse MediaMessage

23.9.6.2 Konstanten der Klasse MediaMessage

Die Klasse MediaMessage verfügt über die folgenden Konstanten:

- AsyncDone, AsyncStart, Buffering, ClockLost, DurationChanged, Element, Eos
- Error, Info, Latency, NewClock, Progress, Qos, RequestState, SegmentDone, Start
- StateAsync, StateChanged, Status, StepDone, StepStart, StreamStart, StreamStatus
- Tag, Toc, Warning

Auf der Webseite <https://gambaswiki.org/wiki/comp/gb.media/mediamessage> finden Sie die zu den Konstanten gehörenden Integer-Werte.

23.9.6.3 Beispiel

Im folgenden Quelltext-Abschnitt werden die Mitteilungen aus dem Datenstrom ausgelesen und in der Konsole der IDE angezeigt:

```
Public Sub hPipeline_Event(Message As MediaMessage)

    Dim vVariant As Variant

    Print "MESSAGE.NAME = "; Message.Name
    Print "-----"
    For Each vVariant In Message
        Print "KEY = "; Message.Key; " -->> VALUE = "; Message[Message.Key]
    Next
    Print "-----"

End
```

Das ist ein Ausschnitt der kontinuierlich ausgelesenen Mitteilungen – ohne Filter:

```
...
-----
MESSAGE.NAME = GstMessageStateChanged
-----
KEY = old-state -->> VALUE = null
KEY = new-state -->> VALUE = ready
KEY = pending-state -->> VALUE = paused
```

```

-----
MESSAGE.NAME = GstMessageNeedContext
-----
KEY = context-type -->> VALUE = gst.soup.session
-----
MESSAGE.NAME = GstMessageHaveContext
-----
KEY = context -->> VALUE =
-----
MESSAGE.NAME = GstMessageStreamStatus
-----
KEY = type -->> VALUE = create
KEY = owner -->> VALUE =
KEY = object -->> VALUE =
-----
MESSAGE.NAME = GstMessageStateChanged
-----
KEY = old-state -->> VALUE = ready
KEY = new-state -->> VALUE = paused
KEY = pending-state -->> VALUE = void-pending
-----
MESSAGE.NAME = GstMessageStreamStatus
-----
KEY = type -->> VALUE = enter
KEY = owner -->> VALUE =
KEY = object -->> VALUE =
-----
MESSAGE.NAME = http-headers
-----
KEY = uri -->> VALUE = http://icecast.ndr.de/ndr/ndr1wellenord/kiel/mp3/128/stream.mp3
KEY = http-status-code -->> VALUE = 200
KEY = redirection-uri -->> VALUE = http://f141.rndfnk.com/ard/ndr/ndr1wellenord/kiel/mp3/128/stream.mp3?
cid=01FBQ1T52FB81EDG99JE36KN5J&sid=2aluMgAc2xEGJVo7hUwbHeTV3gl&token=MUKpI-R3L45S55UKaBf6NJtcAZ8ysRIr3s-
g9wLhh3DY&tvf=6CRHShweqRdmMTQxLnJuZGZuay5jb20
KEY = request-headers -->> VALUE =
KEY = response-headers -->> VALUE =
-----
...

```

Im alternativen Quelltext-Abschnitt wird ein Mitteilungsfiler mit einem frei festgelegten Mitteilungsna-
men eingesetzt → Message.Name = GstMessageStateChanged:

```

Public Sub hPipeline_Event(Message As MediaMessage)

    Dim i As Integer

    If Message.Name = "GstMessageStateChanged" Then
        Print
        Print "MESSAGE.NAME = "; Message.Name
        Print "-----"
    Endif
    For i = 0 To Message.Count - 1
        If Message.Name = "GstMessageStateChanged" Then
            Print "KEY = "; Message.Keys[i]; " -->> VALUE = "; Message[Message.Keys[i]]
        Endif
    Next

End

```

Hier ein Ausschnitt der kontinuierlich ausgelesenen Mitteilungen – mit Filter:

```

...
MESSAGE.NAME = GstMessageStateChanged
-----
KEY = old-state -->> VALUE = null
KEY = new-state -->> VALUE = ready
KEY = pending-state -->> VALUE = paused
-----
MESSAGE.NAME = GstMessageStateChanged
-----
KEY = old-state -->> VALUE = ready
KEY = new-state -->> VALUE = paused
KEY = pending-state -->> VALUE = void-pending
...

```