

Style-Deklaration im HEAD)

Es gilt: Wenn gleiche Bezeichner verwendet, so nur Werte des **zuletzt** geparsten Bezeichners verwendet !!!

Die CLASS-Deklaration aus dem HEAD des Dokumentes wird wertmäßig durch die Style-Deklaration per Attribut des HTML-Elementes überschrieben, wenn gleiche Style-Eigenschaften betroffen sind (ansonsten hinzufügen).

Hinweis zu dynamischen Eigenschaftenveränderung zur Laufzeit:

Die zuletzt während der Laufzeit getätigte Definition ersetzt wertmäßig den aktuellen Attributwert wenn gleiche Attributnamen/Eigenschaften betroffen sind (sonst hinzufügen).

Syntax:

```
objekt.runtimeStyle.xxxxxx
```

mit xxxx als Style-Objekt-Eigenschaft

Beispiel 1:

```
<DIV onclick="this.runtimeStyle.cssText = 'color:red;background-color:blue;border:5px solid black;'">
    Das ist ein DIV. Klick fuer Style-Aenderung.
</DIV>
```

Beispiel 2:

```
<SCRIPT>
function FarbeAendern(FarbeNameAlsKette)
{
    if (ID_DIV.runtimeStyle.backgroundColor != ID_DIV.style.backgroundColor)
    { ID_DIV.runtimeStyle.backgroundColor = FarbeNameAlsKette;}

    alert(    ID_DIV.style.backgroundColor + "\n"
            +    ID_DIV.currentStyle.backgroundColor + "\n"
            + ID_DIV.runtimeStyle.backgroundColor
            );
}
</SCRIPT>
<DIV ID= "ID_DIV">
    Das ist ein DIV
</DIV>
<INPUT TYPE = "button" VALUE = "FarbeAendern" onclick="FarbeAendern('blue')">
```

Eigenschaften:

alle Eigenschaften des style Objektes

.onOffBehavior deprecated ab IE 5.x
Unterstützung von DirectAnimation z.B. für 2D, 3D, Sound

Methoden:

.getAttribute() Wert eines per HTML erzeugten Attributes liefern
DOM nicht geändert

.getExpression() Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern
Style-Eigenschaft ist per Methoden
expression() oder setExpression()
zu definieren
DOM wird nicht verändert (nur Werteveränderung), aber das Dokument-Layout
(nach dem eventuellen expliziten Dokument-Refresh)

.removeAttribute() entfernen eines per HTML erzeugten Attributes
Achtung: Der Browser unterscheidet zwischen HTML-erzeugte oder mit dieser Methode erzeugte Attribute!
per Methode .createAttribute() erzeugte Attribute werden nicht erfasst
DOM wird geändert

.removeExpression() Ausdruck entfernen, der für die Berechnung des Wertes einer Style-Eigenschaft als Objektreferenz der Form objekt.style.eigenschaft. dient.
Ausdruck muss mit der Methode .setExpression() gesetzt worden sein
DOM wird nicht geändert

.setAttribute() Wert von vorhandenem Attribut setzen
wenn Attribut nicht vorhanden, so wird es automatisch erzeugt und mit dem Wert gefüllt
DOM wird nur bei Erzeugung geändert

.setExpression() Wert definieren, der als Ausdruck für die Methode .getExpression() zur Berechnung einer Style-Eigenschaft als Objektreferenz der Form objekt.style.eigenschaft. dient
Ausdruck nur als Script kodierbar
DOM wird nicht geändert

4.3.2.2.4.3.39.23. Standard-Behavior (Verhaltensweisen) von Objekten des Internet Explorer (Auswahl)

Behavior sind erst ab IE 5.x möglich, z.T. erst ab IE 5.5 bzw. IE 6.x..

Das Objekt document.namespaces ist erst ab IE 5.5 implementiert.

Eine Erweiterung der Verhaltensweisen eines Elementes (Objektes) kann auch über extern definierte Anweisungen erfolgen. Diese Anweisungen liegen nicht im Dokument sondern in einer HTML Component (HTC) Datei vom Typ *.htc. Zur Wirksamkeit der Anweisungen muss die Datei in das Dokument geladen werden. Die Verhaltensweise ist also erst nach dem kompletten Laden - inklusive der *.htc-Datei - nutzbar (window.onload Event muss ausgelöst sein).



Eine *.htc-Datei muss aus Anweisungen bestehen, die der Browser interpretieren und ausführen kann, um damit das Verhalten von Elementen zu realisieren. Das gilt vorallem für ein privates Behavior. Siehe unbedingt document.namespace Objekt. Zu Aufbau und Funktionsweise der *.htc-Datei siehe Objekt element.

Ein Behavior kann per STYLE-Attribut, Script oder XML implementiert werden.

In HTML und XML muss ein ID-Attribut kodiert werden !

Der Internet Explorer besitzt Standard-Behavior, die nachfolgend beschrieben werden.

Es ist nicht nötig, eine *.htc-Datei einzubinden, da die StandardBehavior im Browser selbst liegen.

Diese Verhaltensweisen haben Objektcharakter und besitzen z.T. Eigenschaften wie Methoden.

Aus Übersichtsgründen werden Behavior nicht als Objekte beschrieben, da sie aus Sicht von Javascript direkt mit dem style-Objekt in Verbindung stehen.

Beispiele für diverse Kodierungsformen der Standard-Behavior, außer für .style.time2 Behavior: siehe dort

Beispiel für Einbindung eines XML-Namensraumes:

```
<HTML XMLNS:IE>
<HEAD>
<STYLE>
    @media all {IE\:clientCaps {behavior:url(#default#clientCaps)}}
</STYLE>
<SCRIPT>
    function window.onload()
    {
        ID_Pre.innerText =
            "availHeight      = " + ID_ClientCaps.availHeight
            + "\n" + "availWidth    = " + ID_ClientCaps.availWidth
            + "\n" + "bufferDepth  = " + ID_ClientCaps.bufferDepth
            + "\n" + "colorDepth   = " + ID_ClientCaps.colorDepth
            + "\n" + "connectionType = " + ID_ClientCaps.connectionType
            + "\n" + "cookieEnabled = " + ID_ClientCaps.cookieEnabled
            + "\n" + "cpuClass     = " + ID_ClientCaps.cpuClass
            + "\n" + "height       = " + ID_ClientCaps.height
            + "\n" + "javaEnabled  = " + ID_ClientCaps.javaEnabled
            + "\n" + "platform     = " + ID_ClientCaps.platform
            + "\n" + "systemLanguage = " + ID_ClientCaps.systemLanguage
            + "\n" + "userLanguage  = " + ID_ClientCaps.userLanguage
            + "\n" + "width        = " + ID_ClientCaps.width
            + "\n";
    }
</SCRIPT>
</HEAD>
<BODY>
    <IE:clientCaps ID="ID_ClientCaps" >
    <PRE ID="ID_Pre"></PRE>
</BODY>
</HTML>
```

Beispiel für Behavior-Deklaration direkt im STYLE-Attribut:

```
<HEAD>
<SCRIPT>
<!--
    function window.onload() // Achtung: überschreibt Standard-onload-Routine !!
    {
        var Kette = ID_ClientCaps.getComponentVersion(
            "{89820200-ECBD-11CF-8B85-00AA005B4383}",
            "componentid"
        );

        ID_Div.innerHTML = "<FONT SIZE=4>Internet Explorer Version = "
            + Kette
            + "</FONT>";
    }
-->
</SCRIPT>
</HEAD>
<BODY BGCOLOR="#FFFFFF"
    STYLE="behavior:url(#default#clientCaps)" ID="ID_ClientCaps"
>
    <DIV ID="ID_Div"></DIV>
</BODY>
```

Beispiel für XML-Namensraum und STYLE-Kodierung:



```

<HTML XMLNS:IE>
<HEAD>
<SCRIPT>
    function NachDemDownload(DownLoadedFileContent)
    {alert (DownLoadedFileContent);}
</SCRIPT>
</HEAD>
<BODY>
    <IE:Download ID="ID_IETag" STYLE="behavior:url(#default#download)" >
    <P>
    Click
    <A HREF="javascript:ID_IETag.startDownload('download.htm', NachDemDownload)">
    hier
    </A>
    zum Start des Downloads dieser Seite.
</BODY>
</HTML>

```

Beispiel für Behavior-Deklaration durch Zuweisung auf das STYLE-Attribut:

```

<HEAD>
<SCRIPT>
    function window.onload()
    {
        ID_ClientCaps.style.behavior = "url(#default#clientCaps)";

        var DataBindingVerfuegbar = ID_ClientCaps.isComponentInstalled(
            "{9381D8F2-0288-11D0-9501-00AA00B911A5}",
            "componentid"
        );

        if (!DataBindingVerfuegbar)
        {
            ID_ClientCaps.addComponentRequest(
                "{9381D8F2-0288-11D0-9501-00AA00B911A5}",
                "componentid"
            );

            DataBindingVerfuegbar = ID_ClientCaps.doComponentRequest();

            alert(DataBindingVerfuegbar);
        }
    }
</SCRIPT>
</HEAD>
<BODY BGCOLOR="#FFFFFF" ID="ID_ClientCaps">
</BODY>

```

Beispiel für Kodierung per Klasse im HEAD:

```

<HEAD>
<STYLE>
    FolderKlasse {behavior:url(#default#httpFolder);}
</STYLE>
<SCRIPT>
    function Anzeigen()
    {
        var Ordner=location.href.substring(0,location.href.lastIndexOf("/"));
        ID_Span.navigate(Ordner);
    }
</SCRIPT>
</HEAD>
<BODY>
    <SPAN ID="ID_Span" CLASS = "FolderKlasse" onclick = "Anzeigen()">
    </SPAN>
</BODY>

```

4.3.2.2.4.3.39.23.1. .style.clientCaps Behavior des Internet Explorer

verwaltet Informationen vom Internet Explorer unterstützten Standard-Verhaltensarten der Elemente ab IE 5.x

siehe auch document.namespace Objekt

Syntax:

```

XML      <Prefix: CustomTag ID=kette STYLE="behavior:url(#default#clientCaps)" >
HTML     <ELEMENT STYLE="behavior:url(#default#clientCaps)" ID= kette >
Scripting object.style.behavior = "url(#default#clientCaps)"
          object.addBehavior ("#default#clientCaps")

```



Hinweis zur XML-Kodierung:

Anstelle der Tagbegrenzung>

sollte ein eigenständiges Endetag kodiert werden, also

.....>
....
</.....>

Prefix	laut XML-Namensraum
CustomTag	User-definierter Tag
kette	String mit dem ID muss kodiert werden

Beispiel für Eigenschaften:

```
<HTML XMLNS:IE>
<HEAD>
<STYLE>
    @media all {IE\:clientCaps {behavior:url(#default#clientCaps)}}
</STYLE>
<SCRIPT>
    function window.onload()
    {
        ID_Pre.innerText =
            "availHeight"      = " + ID_ClientCaps.availHeight
            + "\n" + "availWidth" = " + ID_ClientCaps.availWidth
            + "\n" + "bufferDepth" = " + ID_ClientCaps.bufferDepth
            + "\n" + "colorDepth" = " + ID_ClientCaps.colorDepth
            + "\n" + "connectionType" = " + ID_ClientCaps.connectionType
            + "\n" + "cookieEnabled" = " + ID_ClientCaps.cookieEnabled
            + "\n" + "cpuClass" = " + ID_ClientCaps.cpuClass
            + "\n" + "height" = " + ID_ClientCaps.height
            + "\n" + "javaEnabled" = " + ID_ClientCaps.javaEnabled
            + "\n" + "platform" = " + ID_ClientCaps.platform
            + "\n" + "systemLanguage" = " + ID_ClientCaps.systemLanguage
            + "\n" + "userLanguage" = " + ID_ClientCaps.userLanguage
            + "\n" + "width" = " + ID_ClientCaps.width
            + "\n";
    }
</SCRIPT>
</HEAD>
<BODY>
    <IE:clientCaps ID="ID_ClientCaps" >
    <PRE ID="ID_Pre"></PRE>
</BODY>
</HTML>
```

Beispiel 1 für Methoden:

```
<HEAD>
<SCRIPT>
<!--
    function window.onload() // Achtung: überschreibt Standard-onload-Routine !!
    {
        var Kette = ID_ClientCaps.getComponentVersion(
            "{89820200-ECBD-11CF-8B85-00AA005B4383}",
            "componentid"
        );

        ID_Div.innerHTML = "<FONT SIZE=4>Internet Explorer Version = "
            + Kette
            + "</FONT>";
    }
-->
</SCRIPT>
</HEAD>
<BODY BGCOLOR="#FFFFFF"
    STYLE="behavior:url(#default#clientCaps)" ID="ID_ClientCaps"
>
    <DIV ID="ID_Div"></DIV>
</BODY>
```

Beispiel 2 für Methoden:



```

<HEAD>
<SCRIPT>
    function window.onload()
    {
        ID_ClientCaps.style.behavior = "url(#default#clientCaps)";

        var DataBindingVerfuegbar = ID_ClientCaps.isComponentInstalled(
            "{9381D8F2-0288-11D0-9501-00AA00B911A5}",
            "componentid"
        );

        if (!DataBindingVerfuegbar)
        {
            ID_ClientCaps.addComponentRequest(
                "{9381D8F2-0288-11D0-9501-00AA00B911A5}",
                "componentid"
            );

            DataBindingVerfuegbar = ID_ClientCaps.doComponentRequest();

            alert(DataBindingVerfuegbar);
        }
    }
</SCRIPT>
</HEAD>
<BODY BGCOLOR="#FFFFFF" ID="ID_ClientCaps">
</BODY>

```

Eigenschaften:

.availHeight	verfügbare Höhe des Arbeitsbereiches auf dem Bildschirm ohne Windows-Taskbar Behavior .style.clientCaps
.availWidth	verfügbare Breite des Arbeitsbereiches auf dem Bildschirm ohne Windows-Taskbar Behavior .style.clientCaps
.bufferDepth	Anzahl der Bits pro Pixel für eine Farbe UND Verwendung des off-screen bitmap buffer Behavior .style.clientCaps
.colorDepth	Anzahl der Bits pro Pixel für eine Farbe UND Verwendung destination device or buffer wird durch den Wert der Eigenschaft bufferDepth überschrieben, wenn .bufferDepth mit Wert > 0 Behavior .style.clientCaps
.connectionType	Typ der genutzten Verbindung bzw. Offline-Status der Verbindung Behavior .style.clientCaps
.cookieEnabled	Cookienutzbarkeit im Browser Behavior .style.clientCaps
.cpuClass	CPU-Hersteller Behavior .style.clientCaps
.height	Auflösung des Bildschirms in Höhe, also Anzahl der vertikalen Pixel Behavior .style.clientCaps
.javaEnabled	Verfügbarkeit der Microsoft Virtual Machine (Microsoft VM) für Java Achtung: Aufgrund eines Rechtsstreites mit dem Unternehmen Sun entwickelt Microsoft ab IE 6.x die MS VM nicht mehr weiter, hält die VM aber noch per Download verfügbar. Für eine aktuellere Version muss der User selbständig beim Unternehmen Sun eine Lizenz für Java erwerben und die zugehörige Software installieren (inklusive Updates). Es reicht dabei die Run-Time-Version der Virtuellen Java Maschine von Sun. Ob allerdings Sun-Java-Standards in Microsoft-Produkten implementiert werden, ist nicht gesichert.
.platform	Behavior .style.clientCaps Windows-Betriebssystem Behavior .style.clientCaps
.systemLanguage	Standardsprache des Betriebssystems Behavior .style.clientCaps
.userLanguage	Sprache des Betriebssystems laut UserEinstellung Behavior .style.clientCaps
.width	Auflösung des Bildschirms in Breite, also Anzahl der horizontalen Pixel Behavior .style.clientCaps

Methoden:

Nachfolgende Methoden behandeln die im IE per Active-X (Active Setup) installierbaren vordefinierten Komponenten. Der Umfang der Komponenten ist abhängig von der IE-Version. Das Installieren der Komponente ist das Binden von deren Daten in den IE (Active Setup). Die Komponenten werden dabei in einem Puffer (Queue) gehalten. Dieser Puffer wird durch den Download der Komponente gefüllt. Es kann nur eine gepufferte Komponente installiert werden.

ID der Komponenten werden in Gross- und Kleinschreibung unterschieden.

Ein ID ist z.B. der String "{9381D8F2-0288-11D0-9501-00AA00B911A5}"

Typ der Komponente ist immer "componentid", wobei Gross-Klein egal ist

Beispiel 1 für Methoden:



```

<BODY BGCOLOR="#FFFFFF"
  STYLE="behavior:url(#default#clientCaps)" ID="ID_ClientCaps"
>
  <DIV ID="ID_Div"></DIV>
  <SCRIPT>
  <!--
    function window.onload()
    {
      var Kette = ID_ClientCaps.getComponentVersion(
        "{89820200-ECBD-11CF-8B85-00AA005B4383}",
        "componentid"
      );

      ID_Div.innerHTML = "<FONT SIZE=4>Internet Explorer Version = "
        + Kette
        + "</FONT>";
    }
  -->
  </SCRIPT>
</BODY>

```

Beispiel 2 für Methoden:

```

<HEAD>
<SCRIPT>
  function window.onload()
  {
    ID_ClientCaps.style.behavior = "url(#default#clientCaps)";

    var DataBindingVerfuegbar = ID_ClientCaps.isComponentInstalled(
      "{9381D8F2-0288-11D0-9501-00AA00B911A5}",
      "componentid"
    );

    if (!DataBindingVerfuegbar)
    {
      ID_ClientCaps.addComponentRequest(
        "{9381D8F2-0288-11D0-9501-00AA00B911A5}",
        "componentid"
      );
      DataBindingVerfuegbar = ID_ClientCaps.doComponentRequest();
      alert(DataBindingVerfuegbar);
    }
  }
</SCRIPT>
</HEAD>
<BODY BGCOLOR="#FFFFFF" ID="ID_ClientCaps">
</BODY>

```

.addComponentRequest()	Komponente zum Download anfordern und nach dem Download installieren Behavior .style.clientCaps
.clearComponentRequest()	Puffer der per .addComponentRequest() angeforderten Downloads löschen Behavior .style.clientCaps
.compareVersions()	2 Versionen einer Komponente vergleichen Bsp. für Version "5,0,18,1024"
.doComponentRequest()	Start des Download aller per .addComponentRequest() angeforderten Komponenten laut Puffer Behavior .style.clientCaps
.getComponentVersion()	Version einer Komponente Behavior .style.clientCaps
.isComponentInstalled()	Verfügbarkeit der Komponente (gedownload UND installiert UND aktuelle Versionsnummer ist mindestens die minimale Versionsnummer) Behavior .style.clientCaps

4.3.2.2.4.3.39.23.2. .style.download Behavior des Internet Explorer

Download einer Datei und Aufruf einer freiprogrammierbaren Funktion direkt nach dem kompletten Ende des Downloads
Behavior erst wirksam nach dem kompletten Laden des Dokumentes (window.onload Event muss ausgelöst worden sein)
Kodierung nur innerhalb derselben Domain (downzuladende Datei muss in selber Domain liegen)

ab IE 5.x

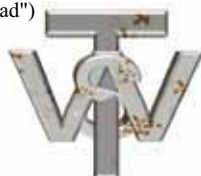
siehe auch document.namespace Objekt

Syntax:

```

XML    <Prefix: CustomTag ID=kette STYLE="behavior:url(#default#download)" >
HTML   <ELEMENT STYLE="behavior:url(#default#download)" ID=kette>
Scripting object.style.behavior = "url(#default#download)"
        object.addBehavior("#default#download")

```



Hinweis zur XML-Kodierung:

Anstelle der Tagbegrenzung>

sollte ein eigenständiges Endetag kodiert werden, also

.....>
....
</.....>

Prefix laut XML-Namensraum
CustomTag User-definierter Tag
kette String mit dem ID
muss kodiert werden

Beispiel:

```
<HTML XMLNS:IE>
<HEAD>
<SCRIPT>
    function NachDemDownload(DownLoadedFileContent)
    {alert (DownLoadedFileContent);}
</SCRIPT>
</HEAD>
<BODY>
    <IE:Download ID="ID_IETag" STYLE="behavior:url(#default#download)" >
    <P>
    Click
    <A HREF="javascript:ID_IETag.startDownload('download.htm', NachDemDownload)">
    hier
    </A>
    zum Start des Downloads dieser Seite.
</BODY>
</HTML>
```

Eigenschaften:

keine

Methoden:

.startDownload() Start des Download

4.3.2.2.4.3.39.23.3. .style.homePage Behavior des Internet Explorer

enthält die Informationen nur für Homepages innerhalb einer gemeinsamen Domain

ab IE 5.x

siehe auch document.namespace Objekt

Syntax:

XML <Prefix: CustomTag ID=kette STYLE="behavior:url('#default#homePage')" >
HTML <ELEMENT STYLE="behavior:url('#default#homePage')" ID=kette>
Scripting object.style.behavior = "url('#default#homePage')"
object.addBehavior ("#default#homePage")

Hinweis zur XML-Kodierung:

Anstelle der Tagbegrenzung>

sollte ein eigenständiges Endetag kodiert werden, also

.....>
....
</.....>

Prefix laut XML-Namensraum
CustomTag User-definierter Tag
kette String mit dem ID
muss kodiert werden

Beispiel

```
<HTML XMLNS:IE>
<HEAD>
    <STYLE>
        @media all {IE\homePage {behavior:url(#default#homepage)}}
    </STYLE>
<SCRIPT>
    function HomepageErmitteln()
    {
        alert(ID_BenutzerTag.isHomePage(ID_Input.value));
        event.returnValue = false;
    }

    function HomepageSetzen()
    {
```



```

        ID_BenutzerTag.setHomePage(ID_Input.value);
        event.returnValue = false;
    }

    function HomepageAnspringen()
    {
        ID_BenutzerTag.navigateHomePage();
        event.returnValue=false;
    }
</SCRIPT>
</HEAD>
<BODY>
    <IE:homePage ID="ID_BenutzerTag" >
    <INPUT TYPE=text ID="ID_Input" VALUE="http://www.test.de ">
    <INPUT TYPE=button VALUE="Homepage ermitteln" onclick="HomepageErmitteln()">
    <INPUT TYPE=button VALUE="Homepage setzen" onclick="HomepageSetzen()">
    <INPUT TYPE=button VALUE=" Homepage anspringen" onclick="HomepageAnspringen()">
</BODY>
</HTML>

```

Beispiel für Bookmark (Favoriten) setzen

```

<a href="#" onClick="this.style.behavior='url(#default#homepage)';
this.setHomePage('http://www.test.de');">Machen Sie test.de zu Ihrer Startseite</a>

```

Eigenschaften:

keine

Methoden:

.isHomePage()	Lage der Homepage auf einer Domain
.navigateHomePage()	Homepage anspringen im Browser
.setHomePage()	Homepage-Lade-Dialogbox aufrufen

4.3.2.2.4.3.39.23.4. .style.httpFolder Behavior des Internet Explorer

Darstellung eines HTTP-Verzeichnis im Fenster

ab IE 5.x

siehe auch document.namespace Objekt

Syntax:

XML	<Prefix: CustomTag ID=kette STYLE="behavior:url('#default#httpFolder')">
HTML	<ELEMENT STYLE="behavior:url('#default#httpFolder')" ID=kette>
Scripting	object.style.behavior = "url('#default#httpFolder')" object.addBehavior ("#default#httpFolder")

Hinweis zur XML-Kodierung:

Anstelle der Tagbegrenzung>

sollte ein eigenständiges Endetag kodiert werden, also

```

.....>
....
</.....>

```

Prefix	laut XML-Namensraum
CustomTag	User-definierter Tag
kette	String mit dem ID muss kodiert werden

Beispiel

```

<HEAD>
<STYLE>
    .FolderKlasse {behavior:url(#default#httpFolder);}
</STYLE>
<SCRIPT>
    function Anzeigen()
    {
        var Ordner=location.href.substring(0,location.href.lastIndexOf("/"));
        ID_Span.navigate(Ordner);
    }
</SCRIPT>
</HEAD>
<BODY>
    <SPAN ID="ID_Span" CLASS = "FolderKlasse" onclick = "Anzeigen()">
    </SPAN>
</BODY>

```

Eigenschaften:

keine

Methoden:

.navigate()	HTTP-Verzeichnis im aktuellen Fenster anzeigen
-------------	--



.navigateFrame() HTTP-Verzeichnis im Fenster nach Wahl anzeigen

4.3.2.2.4.3.39.23.5. .style.mediaBar Behavior des Internet Explorer

Basis-User-Interface für Media-Wiedergabe im Browser-Fenster der Media Bar (Medien-Fenster) mit diversen Funktionalitäten z.B.
 Steuerung der Wiedergabe
 Navigation per HTML z.B. Link
 Playersoftware (Interface) Windows Media Player als Media Bar Player
 laden von HTML in das Medienfenster z.B. per Link, der auf das Ziel "_media" verweisen muss, also das Medien-Fenster

Beispiel für HTML-Link

```
<A HREF="http://www.test.de/test.asf" TARGET="_media"></A>
<A HREF="http://www.test.de/start.htm" TARGET="_media"></A>
```

ab IE 6.x nur unter Windows 32Bit

öffnen der Media Bar durch

- Userklick auf Link im Browserfenster
- Userklick auf Medien-Button in der Menüleiste des IE
- Usereingabe in der Adresszeile einer Url von einer Media-Datei

Es kann zu jedem Zeitpunkt nur genau 1 Media-Datei wiedergegeben werden, es sei denn, man öffnet mehrere Browser-Instanzen.

Die Wiedergabe ist nicht möglich, wenn der User in der Media Bar per Links navigiert, da damit die aktuelle Url in der Media Bar geändert wird gegenüber der Url der gerade wiedergegebenen Media-Datei.

Eine aktive Wiedergabe wird durch HTML-Navigation sofort gestoppt.

Wird die Media-Datei nicht gefunden oder ist der Mime-Typ der Media-Datei unbekannt, dann wird eine Standard-Seite in das Medien-Fenster geladen.

Wird während der HTML-Navigation ein Ziel nicht geladen (weil z.B. nicht existent), dann wird eine Standard-Seite in das Medien-Fenster geladen.

Scriptsteuerung des Behavior erst möglich, wenn window.onload Event erzeugt wurde.

Das Dokument, das die Behavior-Deklaration enthält, muss in die Media Bar geladen werden, damit das Behavior nutzbar ist.

Bsp.:

start.htm enthält die Behavior-Deklaration

Syntax:

```
XML <Prefix: CustomTag ID=kette STYLE="behavior:url(#default#mediaBar)" >
HTML <ELEMENT STYLE="behavior:url(#default#mediaBar)" ID=kette>
Scripting object.style.behavior = "url(#default#mediaBar)"
        object.addBehavior ("#default#mediaBar")
```

Hinweis zur XML-Kodierung:

Anstelle der Tagbegrenzung>

sollte ein eigenständiges Endetag kodiert werden, also

```
.....>
....
</.....>
```

Prefix laut XML-Namensraum
CustomTag User-definierter Tag
kette String mit dem ID
muss kodiert werden

Beispiel:

```
<DIV ID="ID_Div"
STYLE="behavior:url(#default#mediaBar)"
>
</DIV>
<INPUT TYPE=button
VALUE='abspielen von test.asx'
onclick=" ID_Div.playURL('http://www.test.de/test.asx','video/x-ms-asf');
ID_Div.disabledUI = true;
ID_Div.enabled = false;
"
>
```

Eigenschaften:

.currentItem Zeiger auf aktuellen Media-Eintrag (Objekt MediaItem)
 Eintrag stammt aus der Playliste, wenn diese existent ist, und ist der aktuelle Eintrag der Playliste



	siehe Behavior .style.mediaBar
.disabledUI	Sichtbarkeit des User-Interfaces der Media Bar
	siehe Behavior .style.mediaBar
.enabled	Sichtbarkeit des Media Bar Player
	Media Bar Player ist der Windows Media Player
	siehe Behavior .style.mediaBar
.hasNextItem	aktueller Eintrag in der Playliste (Objekt PlaylistInfo) hat einen Nachfolger-Eintrag
	siehe Behavior .style.mediaBar
.nextItem	Zeiger auf nächsten Eintrag in der Playliste (Objekt PlaylistInfo)
	siehe Behavior .style.mediaBar
.openState	Status des Media Bar Player bezüglich Playlist, Codec, Lizenz und Individualisierung
	Media-Datei kann haben
	Codec (Ländernummer)
	Lizenz bei käuflichem Erwerb durch den User
	Individualisierung auf den User
	Media Bar Player ist der Windows Media Player
.playlistInfo	Zeiger auf die Playliste (Objekt PlaylistInfo)
	siehe Behavior .style.mediaBar
.playState	Wiedergabe-Status des Media Bar Player (Wiedergabe der Media-Datei)
	Status wird verändert durch Aktionen des Users mit dem Player
	Media Bar Player ist der Windows Media Player
	Es kann zu jedem Zeitpunkt nur genau 1 Media-Datei wiedergegeben werden, es sei denn, man öffnet mehrere Browser-Instanzen.
	Die Wiedergabe ist nicht möglich, wenn der User in der Media Bar per Links navigiert, da damit die
aktuelle	
	Url in der Media Bar geändert wird gegenüber der Url der gerade wiedergegebenen Media-Datei.
	siehe Methoden .playNext() .playURL() .stop()
	siehe Behavior .style.mediaBar
Methoden:	
.playNext()	Wiedergabe des nächsten Eintrages in der Playliste (Objekt PlaylistInfo) starten
	verändert Eigenschaft .playState
	Es kann zu jedem Zeitpunkt nur genau 1 Media-Datei wiedergegeben werden, es sei denn, man öffnet mehrere Browser-Instanzen.
	Die Wiedergabe ist nicht möglich, wenn der User in der Media Bar per Links navigiert, da damit die
aktuelle	
	Url in der Media Bar geändert wird gegenüber der Url der gerade wiedergegebenen Media-Datei.
	Eine aktive Wiedergabe wird durch HTML-Navigation sofort gestoppt.
	Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende
	Methoden nutzbar:
	.getItemInfo()
	.getAttributeName()
	Eigenschaften nutzbar:
	.attributeCount
	Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.
	siehe Methoden .playURL() .stop() und Eigenschaft .playState
	siehe Behavior .style.mediaBar
.playURL()	Laden einer Media-Datei in die Media Bar und wenn Laden erfolgreich, so Wiedergabe der Media-Daten
	Achtung: Wird die Media-Datei nicht gefunden oder ist der Mime-Typ der Media-Datei unbekannt, dann wird eine Standard-Seite in das Medien-Fenster geladen.
	verändert Eigenschaft .playState
	Es kann zu jedem Zeitpunkt nur genau 1 Media-Datei wiedergegeben werden, es sei denn, man öffnet mehrere Browser-Instanzen.
	Die Wiedergabe ist nicht möglich, wenn der User in der Media Bar per Links navigiert, da damit die
aktuelle	
	Url in der Media Bar geändert wird gegenüber der Url der gerade wiedergegebenen Media-Datei.
	Eine aktive Wiedergabe wird durch HTML-Navigation sofort gestoppt.
	Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende
	Methoden nutzbar:
	.getItemInfo()
	.getAttributeName()
	Eigenschaften nutzbar:
	.attributeCount
	Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.
	siehe Methoden .playNext() .stop() und Eigenschaft .playState
	siehe Behavior .style.mediaBar
.stop()	aktive Wiedergabe stoppen, also beenden
	verändert Eigenschaft .playState
	Es kann zu jedem Zeitpunkt nur genau 1 Media-Datei wiedergegeben werden, es sei denn, man öffnet mehrere Browser-Instanzen.
	Die Wiedergabe ist nicht möglich, wenn der User in der Media Bar per Links navigiert, da damit die
aktuelle	
	Url in der Media Bar geändert wird gegenüber der Url der gerade wiedergegebenen Media-Datei.
	Eine aktive Wiedergabe wird durch HTML-Navigation sofort gestoppt.



Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende

Methoden nutzbar:

.getItemInfo()
.getAttributeName()

Eigenschaften nutzbar:

.attributeCount

Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.

Achtung: Wird die Methode .stop() innerhalb von 10 Sekunden nach dem ERSTEN Aufruf erneut aufgerufen, dann wird eine Standard-Seite in das Medien-Fenster geladen.

siehe Methoden .playURL() .playNext() und Eigenschaft .playState

siehe Behavior .style.mediaBar

Events:

onhide

erzeugt wenn der Media Bar Player gerade unsichtbar gemacht wird (versteckt wird)

siehe Eigenschaft .enabled

entspricht dem Schliessen des Media Bar Player durch den User

Media Bar Player ist der Windows Media Player

siehe Behavior .style.mediaBar

onopenstatechange

erzeugt, wenn Media Bar Player den Status bezüglich Playliste, Codec, Lizenz und

Individualisierung ändert

siehe Eigenschaft .openState

Media Bar Player ist der Windows Media Player

siehe Behavior .style.mediaBar

onplaystatechange

erzeugt, wenn Media Bar Player den Status bezüglich Wiedergabe ändert

siehe Eigenschaft .playState

Media Bar Player ist der Windows Media Player

siehe Behavior .style.mediaBar

onshow

erzeugt wenn der Media Bar Player gerade sichtbar gemacht wird (nicht versteckt wird)

siehe Eigenschaft .enabled

entspricht dem Öffnen des Media Bar Player durch den User

Media Bar Player ist der Windows Media Player

siehe Behavior .style.mediaBar

MediaItem Objekt des Internet Explorer

Repräsentiert für die Media Bar den aktuellen Eintrag in der Playliste (Objekt PlaylistInfo), falls Playliste existent ist

die aktuelle Media-Datei, wenn keine Playliste existent ist

Syntax:

zeiger_auf_mediaBar.MediaItem.eigenschaft

zeiger_auf_mediaBar.MediaItem.methode

zeiger_auf_mediaBar

laut ID-Attribut

Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende

Methoden nutzbar:

.getItemInfo()
.getAttributeName()

Eigenschaften nutzbar:

.attributeCount

Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.

Eigenschaften:

.attributeCount

Anzahl der mit dem MediaItem Objekt verbundenen Attribute liefern

Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende

Methoden nutzbar:

.getItemInfo()
.getAttributeName()

Eigenschaften nutzbar:

.attributeCount

Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.

siehe Behavior .style.mediaBar

.duration

Dauer des MediaItem Objektes in Sekunden

siehe Behavior .style.mediaBar

.name

Name des MediaItem Objektes

siehe Behavior .style.mediaBar

.sourceURL

Url der Media-Datei des MediaItem Objektes

siehe Behavior .style.mediaBar

Methoden:

.getAttributeName()

Name des mit dem MediaItem Objekt verbundenen Attributes liefern

z.B. zur Feststellung, welche Attribute eine Advanced Stream Redirector (ASX)-Datei hat
wie abstract oder author oder copyright

Beispiel für Aufbau eines Eintrages in einer ASX-Datei:

```
<ASX Version="1.0" PreviewMode="No" >
```

```
<entry>
```

```
<title>Testitel</title>
```

```
<author>Testautor</author>
```



```

    <copyright>Test 2002</copyright>
    <abstract>WAV Datei</abstract>
    <ref href=""></ref>
    <banner href = "Testbild.gif" >
        <moreinfo href = "Test.doc"></moreinfo>
        <abstract>besuche www.test.de</abstract>
    </banner>
</entry>
</ASX>

```

Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende

Methoden nutzbar:

.getItemInfo()

.getAttributeName()

Eigenschaften nutzbar:

.attributeCount

Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.

siehe Behavior .style.mediaBar

.getItemInfo()

Wert des mit dem MediaItem Objekt verbundenen Attributes liefern

Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende

Methoden nutzbar:

.getItemInfo()

.getAttributeName()

Eigenschaften nutzbar:

.attributeCount

Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.

siehe Behavior .style.mediaBar

PlaylistInfo Objekt des Internet Explorer:

Repräsentiert die Playlist der Media Bar.

Syntax:

zeiger_auf_mediaBar.PlaylistInfo.eigenschaft

zeiger_auf_mediaBar.PlaylistInfo.methode

zeiger_auf_mediaBar laut ID-Attribut

Eigenschaften:

.attributeCount

Anzahl der mit dem MediaItem Objekt verbundenen Attribute liefern

Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende

Methoden nutzbar:

.getItemInfo()

.getAttributeName()

Eigenschaften nutzbar:

.attributeCount

Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.

siehe Behavior .style.mediaBar

.name

Name des MediaItem Objektes

siehe Behavior .style.mediaBar

Methoden:

.getAttributeName()

Name des mit dem MediaItem Objekt verbundenen Attributes liefern

z.B. zur Feststellung, welche Attribute eine Advanced Stream Redirector (ASX)-Datei hat wie abstract oder author oder copyright

Beispiel für Aufbau eines Eintrages in einer ASX-Datei:

```

<ASX Version="1.0" PreviewMode="No" >
<entry>
    <title>Testitel</title>
    <author>Testautor</author>
    <copyright>Test 2002</copyright>
    <abstract>WAV Datei</abstract>
    <ref href=""></ref>
    <banner href = "Testbild.gif" >
        <moreinfo href = "Test.doc"></moreinfo>
        <abstract>besuche www.test.de</abstract>
    </banner>
</entry>
</ASX>

```

Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende

Methoden nutzbar:

.getItemInfo()

.getAttributeName()

Eigenschaften nutzbar:

.attributeCount



Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.
 siehe Behavior .style.mediaBar
 Wert des mit dem MediaItem Objekt verbundenen Attributes liefern
 Achtung: Nur wenn eine Media-Datei gerade wiedergegeben wird, dann sind per Script folgende Methoden nutzbar:
 .getItemInfo()
 .getAttributeName()
 Eigenschaften nutzbar:
 .attributeCount
 Es ist also vorher die Eigenschaft .playState auf Werte > 10 zu prüfen.
 siehe Behavior .style.mediaBar

4.3.2.2.4.3.39.23.6. .style.saveFavorite Behavior des Internet Explorer

frei definierbare Daten aus dem Dokument sichern bzw. laden:

sichern, wenn **User** das Dokument mit seiner Url als Bookmark zu den Favoriten hinzufügt
 oder User (auch ohne Bookmarksetzung) das Dokument verlässt
 laden, wenn **User** das Dokument aus der Favoritenliste wieder anwählt

ab IE 5.x

siehe auch document.namespace Objekt

Es werden folgende Events vom Objekt **mit** dem saveFavorite-Behavior ausgelöst:

onload	wenn Dokument per Favorit angewählt und somit geladen wird
onsave	wenn Dokument als Bookmark in die Favoritenliste eingetragen wird
oder	wenn User das Dokument verlässt (ohne Bookmarksetzung)

Gespeichert wird immer permanent, also auch über mehrere Online-Sitzungen, also
 in den browserinternen Datenbereich
 und zugleich physisch auf der Festplatte des Users als INI-Datei.
 Pro Objekt wird dazu ein objekteneigener User-Data-Store verwendet.

Bsp. für Aufbau der url-Datei in der Favoritenliste:

```
[DEFAULT]
BASEURL=http://www.test.de/
[InternetShortcut]
URL=http://www.test.de/
Modified=009ECB4522B6C10148
IconFile=http://www.test.de/favicon.ico
IconIndex=1
```

Der Eintrag Modified=009ECB4522B6C10148 wird vom Browser festgelegt, so bald die Bookmark erzeugt wird, und ist somit Bookmark-spezifisch.

Syntax:

XML	<Prefix: CustomTag ID=kette STYLE="behavior:url('#default#saveFavorite')" >
HTML	<ELEMENT STYLE="behavior:url('#default#saveFavorite')" ID=kette>
Scripting	object.style.behavior = "url('#default#saveFavorite')" object.addBehavior("#default#saveFavorite")

Hinweis zur XML-Kodierung:

Anstelle der Tagbegrenzung>

sollte ein eigenständiges Endetag kodiert werden, also

```
.....>
....
</....>
```

Prefix	laut XML-Namensraum
CustomTag	User-definierter Tag
kette	String mit dem ID muss kodiert werden

Beispiel

```
<HTML>
<HEAD>
  <STYLE>
    .FavoriteKlasse {behavior:url('#default#saveFavorite');}
  </STYLE>
  <SCRIPT>
    function InterneDateSpeichern()
    {
      // Textwert des Input als neues Attribut erzeugen und speichern
      ID_Input.setAttribute("InterneDateAlsAttribut", ID_Input.value);
    }
  </SCRIPT>
```



```

function InterneDateLaden()
{
    // Textwert des Input laden
    ID_Input.value= ID_Input.getAttribute("InterneDateAlsAttribut");
}
</SCRIPT>
</HEAD>
<BODY>
<INPUT TYPE=text
        ID="ID_Input"
        CLASS= FavoriteKlasse
        onsave="InterneDateSpeichern()"
        onload="InterneDateLaden()"
>
</BODY>
</HTML>

```

Eigenschaften:

.XMLDocument Referenz auf XML-Dokument (XML-DOM)

Methoden:

.getAttribute() Wert eines per HTML erzeugten Attributes liefern
DOM nicht geändert

.removeAttribute() entfernen eines per HTML erzeugten Attributes
Achtung: Der Browser unterscheidet zwischen HTML-erzeugte oder mit dieser Methode erzeugte Attribute!
per Methode .createAttribute() erzeugte Attribute werden nicht erfasst
DOM wird geändert

.setAttribute() Wert von vorhandenem Attribut setzen
wenn Attribut nicht vorhanden, so wird es automatisch erzeugt und mit dem Wert gefüllt
DOM wird nur bei Erzeugung geändert

4.3.2.2.4.3.39.23.7. .style.saveHistory Behavior des Internet Explorer

frei definierbare Daten aus dem Dokument sichern bzw. laden:

sichern, wenn User das Dokument verlässt

laden, wenn User das Dokument wieder anwählt per Zurück- oder Vorwärts-Button im Browser-Menü

ab IE 5.x

siehe auch document.namespace Objekt

Es werden folgende Events vom Objekt **mit** dem saveHistory-Behavior ausgelöst:

onload	wenn Dokument geladen wird per Zurück- oder Vorwärts-Button im Browser-Menü
onsave	wenn Dokument verlassen wird

Gespeichert wird nicht permanent, also auch nicht über mehrere Online-Sitzungen.

Pro Objekt wird dazu ein objekt eigener User-Data-Store verwendet.

Syntax:

```
<META NAME="save" CONTENT="history">
```

sowie

XML	<Prefix: CustomTag ID=kette STYLE="behavior:url('#default#saveHistory ')" >
HTML	<ELEMENT STYLE="behavior:url('#default#saveHistory')" ID=kette>
Scripting	object.style.behavior = "url('#default#saveHistory')"
	object.addBehavior ("#default#saveHistory")

Hinweis zur XML-Kodierung:

Anstelle der Tagbegrenzung>

sollte ein eigenständiges Endetag kodiert werden, also

```

.....>
....
</.....>

```

Prefix	laut XML-Namensraum
CustomTag	User-definierter Tag
kette	String mit dem ID muss kodiert werden

Die META-Angabe **muss** kodiert werden (im HEAD) !

Beispiel:

```

<HTML>
<HEAD>
  <META NAME="save" CONTENT="history">
  <STYLE>
    .HistoryKlasse {behavior:url('#default#saveHistory');}
  </STYLE>

```



```

<SCRIPT>
    function InterneDateSpeichern()
    {
        // Textwert des Input als neues Attribut erzeugen und speichern
        ID_Input.setAttribute("InterneDateAlsAttribut", ID_Input.value);
    }

    function InterneDateLaden()
    {
        // Textwert des Input laden
        ID_Input.value= ID_Input.getAttribute("InterneDateAlsAttribut");
    }
</SCRIPT>
</HEAD>
<BODY>
    <A HREF="www.test.de">
        Dokument verlassen
    </A>
    <INPUT TYPE=text
        ID="ID_Input"
        CLASS= FavoriteKlasse
        onsave="InterneDateSpeichern()"
        onload="InterneDateLaden()"
    >
</BODY>
</HTML>

```

Eigenschaften:

.XMLDocument Referenz auf XML-Dokument (XML-DOM)

Methoden:

.getAttribute() Wert eines per HTML erzeugten Attributes liefern
DOM nicht geändert

.removeAttribute() entfernen eines per HTML erzeugten Attributes
Achtung: Der Browser unterscheidet zwischen HTML-erzeugte oder mit dieser Methode erzeugte Attribute!
per Methode .createAttribute() erzeugte Attribute werden nicht erfasst
DOM wird geändert

.setAttribute() Wert von vorhandenem Attribut setzen
wenn Attribut nicht vorhanden, so wird es automatisch erzeugt und mit dem Wert gefüllt
DOM wird nur bei Erzeugung geändert

4.3.2.2.4.3.39.23.8. .style.saveSnapshot Behavior des Internet Explorer

frei definierbare Daten aus dem Dokument sichern , wenn User das Dokument per Menüpunkt "Datei-Speichern (unter) " als **HTML-Datei** auf die Festplatte ablegt, also einen HTML-Schnappschuss des Dokumentes erzeugt.

ab IE 5.x

siehe auch document.namespace Objekt

Es wird folgendes Event vom Objekt **mit** dem saveSnapshot-Behavior ausgelöst:

onsave wenn Dokument gespeichert wird auf Festplatte als HTML-Datei

Gespeichert wird nicht permanent, also auch nicht über mehrere Online-Sitzungen.

Pro Objekt wird dazu ein objekteigener User-Data-Store verwendet.

Gespeichert werden können folgende Objekte **nicht:** body Objekt
alle Tabellenelemente (Objekte), die unterhalb dem table Objekt liegen
wie Zelle oder Spalte

Gespeichert werden können **nur** folgende Datentypen:

String, Boolean und Integer-Varianten

also **keine** Zeiger, Felder als Ganzheit (elementweise aber speicherbar wenn Typ wie oben) etc.

Syntax:

<META NAME="save" CONTENT="snapshot">

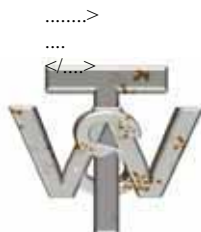
sowie

XML <Prefix: CustomTag ID=kette STYLE="behavior:url('#default#saveSnapshot')">
HTML <ELEMENT STYLE="behavior:url('#default#saveSnapshot')" ID=kette>
Scripting object.style.behavior = "url('#default#saveSnapshot')"
object.addBehavior ("#default#saveSnapshot")

Hinweis zur XML-Kodierung:

Anstelle der Tagbegrenzung>
.....</>

sollte ein eigenständiges Endetag kodiert werden, also



Prefix	laut XML-Namensraum
CustomTag	User-definierter Tag
kette	String mit dem ID muss kodiert werden

Die META-Angabe **muss** kodiert werden (im HEAD) !

Beispiel:

```
<HTML>
<HEAD>
  <META NAME="save" CONTENT="snapshot">
  <STYLE>
    .DateiSpeichernUnterKlasse {behavior:url(#default#saveSnapshot);}
  </STYLE>
  <SCRIPT>
    function InterneDateSpeichern()
    {
      // Textwert des Input als neues Attribut erzeugen und speichern
      ID_Input.setAttribute("InterneDateAlsAttribut", ID_Input.value);
    }
  </SCRIPT>
</HEAD>
<BODY>
  <INPUT TYPE=text
    ID="ID_Input"
    CLASS= .DateiSpeichernUnterKlasse
    onsave="InterneDateSpeichern()"
  >
</BODY>
</HTML>
```

Eigenschaften:

keine

Methoden:

keine

4.3.2.2.4.3.39.23.9. .style.time2 Behavior des Internet Explorer

verwaltet die Zeitsteuerung (Timeline, Zeitlinie) für Elemente eines HTML-Dokumentes wie z.B.

Sound,
Videos
Nicht-Media-Objekten
in Verbindung mit transitionFilter (siehe Objekt filter)

anhand von vordefinierten Time-Containern als Timer, die mehrere Elemente im Dokument (Elemente-Gruppe) steuern können
(Damit entfällt die Programmierung von Timern von Hand per Javascriptmethoden wie .setTimeout() etc.)

anhand beliebig vieler Timer im Dokument

anhand von vordefinierten Time-Container zur Art der Animation von Elementgruppen z.B. sequentiell oder parallel

anhand von vordefinierten Eigenschaften des time2 Behavior zur Vor – und Rückwärtsanimation eines Elementes
auf der Timeline.

anhand von XML-Komponenten (vordefinierte Tags im Format **t:vordefinierter_name_des_container**)

komplett erst ab IE 6.x unterstützt

teilweise ab IE 5.5 unterstützt

Behavior time2 ersetzt Behavior time, weil das time deprecated ist.
benötigt keine *.htc-Datei (HTC-Datei) da im Browser als Standard-Behavior implementiert
erst mit kompletten Laden des Dokumentes verwendbar (window.onload Event muss ausgelöst sein).

Die Unterstützung von Eigenschaften und Methoden eines Objektes bezüglich der Timeline-Steuerung ist objektspezifisch (siehe jeweilige Beschreibung der Objekte).

siehe auch timeChildren Collection und document.body.timeAll Collection

4.3.2.2.4.3.39.23.9.1. Timer Konzept des Internet Explorer

4.3.2.2.4.3.39.23.9.1.1. Ansatz

Anhand der Timed Interactive Multimedia Extensions (zeitgesteuerte interaktive Multimedia Erweiterungen)
wird in HTML die zeitliche Synchronisation von Media- und Nichtmedia-Objekten möglich (z.B. Sound, Video, DIV-Objekt).

Die Time-Gruppierung von HTML-Elementen im IE erfolgt per XML-Tags (siehe Objekt transitionFilter) oder der dem .timeContainer Attribut des .style.time2 Behavior. Basissprache ist Synchronized Multimedia Integration Language (SMIL) 2.0.

Timer sind verschachtelbar.

Zur Zeitsteuerung innerhalb der Gruppe wird die Zeitlinie (Timeline) des jeweiligen Objektes benutzt, die auch vererbt wird. Die Timeline ist somit die Abfolge von Zeitpunkten der Animation des jeweiligen Objektes.



Beispiel: Die Timeline des Dokumentes startet, sobald das Dokument geladen wurde. Das Anzeigen (Rendern) der Dokument-Elemente erfolgt ebenfalls in der Timeline. Diese Timeline kann mit denen der Objekte im Dokument synchronisiert werden.

Zur Synchronisation der Timelines in der Gruppe und zur Interaktion zwischen den Gruppenobjekten werden Events genutzt. Vererbung und Events ermöglichen die zeitliche Synchronisation der Timelines abhängiger Objekte. Nicht jedes Event wird von einem Kind an die Eltern durchgereicht.

Die Interaktion mit dem User ist anhand der Timeline ebenfalls möglich.

Die Timeline einer Media-Datei, die erst gedownloadet werden muss, kann erst **nach** dem kompletten Download aktiv werden. Das Ende des Downloads wird vom time2 Behavior nicht erkannt, muss also programmiert werden per Eventereignis (siehe auch Behavior `.style.download`).

Die Timeline des Dokumentes ist die Timeline des `document.body` Objektes.

Die Unterstützung von Eigenschaften und Methoden eines Objektes bezüglich der Timeline-Steuerung ist objektspezifisch (siehe jeweilige Beschreibung der Objekte).

Die Timeline kann verschiedene Zustände haben:

unbekannt, inaktiv, aktiv	
cueing	Timeline verarbeitet media file
	Hinweis: Element ist aktiv
	nur für Element das media file verarbeiten kann
seeking	Timeline sucht einen Punkt im media file
	Hinweis: Element ist aktiv
	nur für Element das media file verarbeitet
holding	Timeline ist gehalten
	Hinweis: Element ist nicht aktiv, aber Element wartet auf Ende der Timeline der Eltern

Ein Objekt (Element) kann auf seiner Timeline verschiedene Zustände bezüglich der Eventverarbeitung haben:

aktiv	Element kann Events verarbeiten
inaktiv	ohne holding Timeline: Element kann keine Events verarbeiten
inaktiv	bei holding Timeline: Element kann nur das folgende Event verarbeiten:
	Warten auf das Ende der Timeline der Eltern

Es ist möglich, ein Element auf der Timeline rückwärts zu animieren, aber rückwärts nur bezüglich Timeline.

Ein Video-Element rückwärts abzuspielen, setzt voraus, dass frameweise auf der Timeline animiert wird, also die Timeline die Framefolge bestimmt.

Typisches Beispiel für Rückwärtsanimation ist z.B. eine Gruppe von DIV's innerhalb t:SEQ (sequentieller Time-Container), deren sequentielle Reihenfolge der Anzeige einmal vorwärts oder rückwärts sein kann (vorwärts entspricht Kodierungsfolge der DIV's innerhalb des Time-Containers).

4.3.2.2.4.3.39.23.9.1.2. Alternative zum Timer Konzept des IE

Alternativ zum Time-Konzept kann die selbstprogrammierte Zeitsynchronisation verwendet werden, die **nicht nur** dem IE vorbehalten ist. Dazu wird im Dokument als "Herz" eine zentrale Timer-Funktion implementiert, innerhalb derer die Rückkehrcode der Objekte des Dokumentes ausgewertet werden. Anhand dieser Code ist Synchronisation möglich. Die Objekte können selbst Timer-Funktionen haben, müssen aber ihre Ereignisse dem "Herz" melden, das mit dem Laden des Dokumentes aktiviert werden muss. Als Ereignisse werden globale Variablen verwendet, deren Belegung auf einen definierten Wert durch das "Herz" periodisch abgefragt werden. Das "Herz" muss damit eine rekursive Funktion sein, die sich selbst im Zeitintervall aufruft.

Man beachte: Je mehr Timer implementiert sind, um so ungenauer geht die PC-Uhr, da Windows selbst Timer benutzt und diese mit den Dokument-Timern teilen muss. Die PC-Hardware bietet nur begrenzt Timer an, die von der Software benutzt werden.

4.3.2.2.4.3.39.23.9.1.3. XML-Kodierung in HTML

Nachfolgende Beschreibung hat Einführungscharakter und ist nicht umfassend. Die umfassenden konkreten Kodierungsformen eines Objektes zum `.style.time2` Behavior sind der jeweiligen Objektbeschreibung zu entnehmen. Die konkrete Kodierung zum `.style.time2` Behavior ist aus dessen Beschreibung zu entnehmen.

Namensraum:

Hinweis: Nachfolgend fett dargestellte Kodierungsteile sind Pflichtkodierung, also nicht zu verändern.

erzeugen und importieren

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
  <?IMPORT          namespace="t"
                    implementation="#default#time2"
  >
  oder
  <?IMPORT          namespace = t
                    urn = "urn:schemas-microsoft-com:time"
                    implementation = "#default#time2"
  >
  .....
</HEAD>
```



referenzieren

```

<html_element_tag STYLE="behavior:url(#default#time2)" ...>

oder

<HTML>
<HEAD>
  <STYLE>
    .freier_klassen_name {behavior: url(#default#time2);}
  </STYLE>
</HEAD>
<BODY>
  ....
  <html_element_tag CLASS="freier_klassen_name" ...>
</BODY>
</HTML>

```

erzeugen, importieren und referenzieren

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
</HEAD>
<BODY>
  ....
  <html_element_tag STYLE="behavior:url(#default#time2)" ...>
  ....
</BODY>

oder

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
  .freier_klassen_name { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
  ....
  <html_element_tag CLASS="freier_klassen_name" ...>
  ....
</BODY>

```

XML-Tag im Namensraum des Behavior:

sind vordefinierte Tags vorhanden (haben immer Endetag), die eine Timerart repräsentieren.

Bsp:

```

t:EXCL
t:SEQ
t:PAR
t:AUDIO
t:VIDEO

```

Der Namensraum **t** muss erzeugt sein und wird durch den Buchstaben vor dem Doppelpunkt referenziert.

XML-Tags des Behavior sind z.T. verschachtelbar.

4.3.2.2.4.3.39.23.9.1.4. Timeline für ein Objekt anhand HTML-Attribute erzeugen (BEGIN, END, DUR, TIMEACTION)

Objekt kann HTML- oder XML-Element sein

Attribute stammen aus dem Behavior style.time2 (siehe dort, z.T. auch in der Beschreibung anderer Objekte mit genannt z.B. DIV-Objekt)

Beginn der Timeline:

Attribut BEGIN
 Wert in Sekunden als String
 Wartezeit nach dem Laden des Elternobjektes

Ende der Timeline:

Attribut END oder Attribut DUR (Dauer)
 Wert in Sekunden als String
 für END: absoluter Zeitpunkt
 für DUR: relativ zu BEGIN

Aktion während der Timeline:

Attribut TIMEACTION mit Zeichenkettenwert
 Wert abhängig vom Element
 oder auf "none" gesetzt, so trotzdem in den Kinder definierte Aktionen
 ausführbar, wenn Eltern-Timeline aktiv ist
 nur von einem aktiven Element ausführbar und wenn Eltern- und Element-Timeline aktiv sind

Beispiel für HTML-Elemente:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
  <P      ID="ID_P1"
        CLASS="time-line_klasse"
        BEGIN="2"
        DUR="5"
  >
    Test
  </P>
  <P      ID="ID_P2"
        CLASS="time_line_klasse"
        BEGIN="3"
        DUR="6"
        TIMEACTION="display"
  >
    Test
  </P>
  <P      ID="ID_P3"
        CLASS="time_line_klasse"
        BEGIN="4" DUR="7"
        TIMEACTION="visibility"
  >
    Test
  </P>
  <P      ID="ID_P4"
        CLASS="time_line_klasse"
        BEGIN="1"
        DUR="3"
        TIMEACTION="hidden"
  >
    Test
  </P>
  <P      ID="ID_P5"
        CLASS="time_line_klasse"
        BEGIN="1"
        DUR="3"
        TIMEACTION="none"
  >
    Test
  </P>
  <H1     ID="ID_H1"
        CLASS="time_line_klasse"
        BEGIN="0"
        DUR="11"
        TIMEACTION="style"
        STYLE="Color:Red;"
  >
    Test
  </H1>
</BODY>
</HTML>
```

4.3.2.2.4.3.39.23.9.1.5. *Timeline für ein Objektgruppe (Timecontainer) erzeugen (Attribut TIMECONTAINER)*

Objekt kann HTML- oder XML-Element sein (siehe Timeline für Objekt)

Die Objektgruppe wird als Time Container bezeichnet.

Innerhalb des Containers können Objekte z.B. wie folgt synchronisiert werden:

t:EXCL	exklusiv	Timeline der Elemente nicht synchronisieren
t:SEQ	sequentiell	Timeline der Element als Gesamtfolge aus 1 sequentiellen
		Elemente-Animation



t:PAR parallel

Timeline der Element als Gesamtfolge aus parallelen
Elemente-Animationen

Attribute der Objektgruppe:
wie die eines Objektes

Anstelle des Attributes TIMECONTAINER können vordefinierte XML-Tags verwendet werden

z.B. t:SEQ
t:PAR
t:EXCL

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<?IMPORT namespace="t" implementation="#default#time2">
<HEAD>
<STYLE>
.time_line_klasse {behavior: url(#default#time2);}
</STYLE>
</HEAD>
<BODY>
<t:SEQ ID="ID_Seq">
<DIV ID="ID_Div1" CLASS="time_line_klasse" DUR="2">Div1</DIV>
<DIV ID="ID_Div2" CLASS="time_line_klasse" DUR="2">Div2</DIV>
<DIV ID="ID_Div3" CLASS="time_line_klasse" DUR="2">Div3</DIV>
<DIV ID="ID_Div4" CLASS="time_line_klasse">Div4</DIV>
</t:SEQ>
</BODY>
</HTML>
```

4.3.2.2.4.3.39.23.9.1.5.1. Timeline für eine Objektgruppe aus einem unverschachtelten Timer

Das Attribut TIMECONTAINER referenziert den Timer-Typ.

Beispiel 1 Marquee aus Bildern

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2);}
</STYLE>
</HEAD>
<BODY>
<MARQUEE ID="ID_Marquee"
CLASS="time_line_klasse"
TIMECONTAINER="seq"
REPEATCOUNT="indefinite"
>
<IMG ID="ID_Img1" CLASS="time" DUR="4" SRC="test1.gif" ALT="Test1">
<IMG ID="ID_Img2" CLASS="time" DUR="4" SRC="test2.gif" ALT="Test2">
<IMG ID="ID_Img3" CLASS="time" DUR="4" SRC="test3.gif" ALT="Test3">
</MARQUEE>
</BODY>
</HTML>
```

Beispiel 2 Tabelle animieren:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
.TextAufHighLight{ color:#CCCCC; font-weight:bold;}
</STYLE>
<SCRIPT>
function PauseAufheben()
{
ID_Table.resumeElement();
ID_Button1.disabled = false;
ID_Button2.disabled = true;
}

function Pausieren()
{
ID_Table.pauseElement();
ID_Button1.disabled = true;
```



```

        ID_Button2.disabled = false;
    }
</SCRIPT>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
    <TABLE ID="ID_Table"
        BORDER="1"
        CLASS="time_line_klasse"
        DUR="8s"
        REPEATCOUNT="indefinite"
        TIMECONTAINER="SEQ"
        TIMEACTION="none"
    >
    <TBODY>
        <TR>
            <TH WIDTH="120">Eins</TH>
            <TH WIDTH="50">Zwei</TH>
            <TH WIDTH="40">Drei</TH>
        </TR>
        <TR
            ID="ID_TR1"
            CLASS="time_line_klasse"
            TIMEACTION="class: TextAufHighLight "
            DUR="2s"
        >
            <TD>EinsA</TD>
            <TD>ZweiA</TD>
            <TD>DreiA</TD>
        </TR>
        <TR
            ID="ID_TR2"
            CLASS="time_line_klasse"
            TIMEACTION="class: TextAufHighLight "
            DUR="2s"
        >
            <TD>EinsB</TD>
            <TD>ZweiB</TD>
            <TD>DreiB</TD>
        </TR>
    </TBODY>
</TABLE>
<BR>
<BUTTON
    ID="ID_Button1"
    onclick="Pausieren();"
>
    pausieren
</BUTTON>
<BUTTON
    ID="ID_Button2"
    DISABLED="true"
    onclick="PauseAufheben();"
>
    Pause aufheben
</BUTTON>
</BODY>
</HTML>

```

4.3.2.2.4.3.39.23.9.1.5.2. Timeline für eine Objektgruppe aus verschachtelten Timern

Ziel von verschachtelten Timern ist, dass die Kindertimer das Element des Eltern-Timer **ebenfalls** animieren. Die Kindertimer müssen also zum Eltern-Timer passen. Ideal dafür sind z.B. Filter, die als Kinder ein Bild im Eltern-Timer mitanimieren.

Das Attribut TIMECONTAINER referenziert den Timer-Typ.

Das Attribut TIMECONTAINER **muss** in **jedem** Timer, der einen Eltern-Timer hat, kodiert werden, damit der Elterntimer weiss, wie die Kinder das Elternelement mitanimieren. Das ist nötig, um verschachtelte Timer synchronisieren zu können. Der Wert des Attributes im Kind-Timer, also der Typ der Animation per Kind-Timer, muss zur Animation per Eltern-Timer passen. Dafür muss der Programmierer sorgen.

Beispiel 1: Im Beispiel sind die STYLE-Attributangaben zur Position (z.B. left etc.) nur aus Übersichtlichkeit nicht kodiert worden.

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>

```



```

<DIV ID="ID_Div" STYLE="height:100px">
<t:SEQ ID="ID_Seq"
  REPEATCOUNT="indefinite"
>
  <t:MEDIA ID="ID_Media1"
    SRC = "test1.jpg"
    STYLE="position:absolute; ....."
    DUR="3"
    TIMECONTAINER="par"
    FILL="transition"
  >
    <t:TRANSITIONFILTER ID="ID_Transfilter1"
      TYPE="fade"
      DUR="2"
    >
    </t:TRANSITIONFILTER>
    <t:TRANSITIONFILTER ID="ID_Transfilter2"
      TYPE="ClockWipe"
      DUR="2"
    >
    </t:TRANSITIONFILTER>
  </t:MEDIA>
  <t:MEDIA ID="ID_Media2"
    SRC = "test2.jpg"
    STYLE="position:absolute; ....."
    DUR="3"
    TIMECONTAINER="par"
    FILL="transition"
  >
    <t:TRANSITIONFILTER ID="ID_Transfilter3"
      TYPE="ClockWipe"
      DUR="2"
    >
    </t:TRANSITIONFILTER>
    <t:TRANSITIONFILTER ID="ID_Transfilter4"
      TYPE="fade"
      DUR="2"
    >
    </t:TRANSITIONFILTER>
  </t:MEDIA>
</t:SEQ>
</DIV>
</BODY>
</HTML>

```

Beispiel 2: Ein Marquee-Objekt als Kindtimer neben parallelen Filtern (ob Marquee und Filter zueinander und zum Eltern-Timer passen ?)
 Im Beispiel sind die STYLE-Attributangaben zur Position (z.B. left etc.) nur aus Übersichtlichkeit nicht kodiert worden.

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<DIV ID="ID_Div" STYLE="height:100px">
<t:SEQ ID="ID_Seq"
  REPEATCOUNT="indefinite"
>
  <t:MEDIA ID="ID_Media1"
    SRC = "test1.jpg"
    ALT="Test1"
    STYLE="position:absolute; ...."
    DUR="3"
    TIMECONTAINER="par"
    FILL="transition"
  >
    <t:TRANSITIONFILTER ID="ID_Transfilter1"

```



```

        TYPE="fade"
        DUR="2"
    >
</t:TRANSITIONFILTER>

<t:TRANSITIONFILTER      ID="ID_Transfilter2"
        TYPE="ClockWipe"
        DUR="2"
    >
</t:TRANSITIONFILTER>

<MARQUEE      ID="ID_Marquee"
        CLASS="time_line_klasse"
        TIMECONTAINER="seq"
        REPEATCOUNT="indefinite"
        STYLE="position:absolute; ...."
    >
        <IMG      ID="ID_Img1"
                SRC="test2.gif"
                CLASS="time"
                DUR="4"
                ALT="Test2"
            >
        <IMG      ID="ID_Img2"
                SRC="test3.gif"
                CLASS="time"
                DUR="4"
                ALT="Test3"
            >
        <IMG      ID="ID_Img3"
                SRC="test4.gif"
                CLASS="time"
                DUR="4"
                ALT="Test4"
            >
        <IMG      ID="ID_Img4"
                SRC="test5.gif"
                CLASS="time"
                DUR="4"
                ALT="Test5"
            >
    </MARQUEE>
</t:MEDIA>
</t:SEQ>
</DIV>
</BODY>
</HTML>

```

4.3.2.2.4.3.39.23.9.1.6. *Time Formate zum .style.time2 Behavior des Internet Explorer*
für Eigenschaften wie z.B. .begin .dur .end .repeatDur und .syncTolerance

"h:m:s.f"	h	Stunde	als Integer
	m	Minute	als Integer
	s	Sekunde	als Integer oder Float
	f	Hundertstel Sekunden	
jedes Nachfolge-Element kann entfallen			
Beispiele	"25:45:10"	25 Stunden, 45 Minuten, 10 Sekunden	
	"45:35"	45 Minuten, 35 Sekunden	
	"45:00.275"	45 Minuten, 0,275 Sekunde	
	"10.5"	10,5 Sekunden	

"xh" oder "xm" oder "xs" oder "xms"	xh	Stunde	mit x Integer
	xm	Minute	mit x Integer
	xs	Sekunde	mit x als Integer oder Float
	xms	Millisekunde	mit x als Integer

"jjjj-mo-ddThh:mm:ss+hh:mm.TZD"

jjjj	Jahr	als Integer	4 stellig
mo	Monat	als Integer	2 stellig, mit Vornull
dd	Tag	als Integer	2 stellig, mit Vornull
hh	Stunde	als Integer	2 stellig, mit Vornull, 00 bis 23
mm	Minute	als Integer	2 stellig, mit Vornull, 00 bis 59
ss	Sekunde	als Integer oder Float	ohne Vornull



T Trenner zwischen Datum und Uhrzeit
+hh:mm Offset der Stunde und Minute zur Zeit laut TZD
.TZD optional, Platzhalter für Zeitzone, Standard ist UTC (Weltzeit)
 z.B. UTC oder GMT

nur für Eigenschaften .begin und .end kodierbar

objekt_zeiger.event solange Warten bis Ereignis laut event für das Objekt laut id eintritt und dann das Objekt starten
 event entspricht dem Eventbezeichner on "on"
 Bsp: anstelle onclick nur click kodieren
 onbegin nur begin kodieren
 onend nur end kodieren

nur für Eigenschaften .begin und .end kodierbar

id.event+zeit_wert_als_string zeit_wert_als_string siehe oben
 solange Warten bis Ereignis laut event für das Objekt laut id eintritt **und** dann die Wartezeit abwarten
 und dann das Objekt starten
 Standard ist "+0" in Sekunden und kein Event

nur für Eigenschaften .begin und .end kodierbar

Beispiele für Kodierung von Time in der Eigenschaft .begin:

```
object.begin=" objekt_zeiger.begin+10s" // warten bis Event "onbegin" zum Objekt laut
//                                objekt_zeiger (z.B. laut ID-Attribut)
//                                eintritt,
//                                dann 10 Sekunden warten
//                                dann Objekt laut object starten
object.begin=" objekt_zeiger.focus+10s" // warten bis Event "onfocus" zum Objekt laut
//                                objekt_zeiger (z.B. laut ID-Attribut)
//                                eintritt,
//                                dann 10 Sekunden warten
//                                dann Objekt laut object starten

object.begin="2; objekt_zeiger.click+1" // 2 Sekunden nach dem Laden des Elternobjektes von
//                                object warten
//                                dann auf das Ereignis click zum Objekt laut
//                                objekt_zeiger warten
//                                dann 1 Sekunde warten
//                                dann Objekt laut object starten
```

Hinweis: object laut ID-Attribut des Behavior-Objektes von .style.time2.

4.3.2.2.4.3.39.23.9.1.7. Alternativen zum Behavior .style.time2

Alternativen sind: Objekt bgound
 Windows Media Player ab 7.1 (Hinweis: Versionen vor 7.1 sind z.T. sehr inkompatibel zur Version 7.1)

Vergleich .style.time2 Behavior und Windows Media Player ab 7.1:

Die Objekte und Collectionen zum Player und zum Behavior .style.time2 sind konzeptionell ähnlich.
 Der Player und .style.time2 basieren auf gemeinsamen Prinzipien, nur dass sich die referenzierten Objekte und Collectionen anders nennen bzw. verschieden aufgebaut sind.
 Die Eigenschaften zum .style.time2 Behavior und zum Windows Media Player sind sehr ähnlich, wenn nicht gar identisch.

Während der Windows Media Player alle Windows-kompatiblen Medienarten unter einem gemeinsamen Dach verwalten kann, muss beim .style.time2 Behavior das richtige Behavior-Objekt gewählt werden, um entsprechende spezielle Methoden zum Typ des Mediums ansprechen zu können.

Das Objektmodell des Windows Media Players ist wesentlich ganzheitlicher, dadurch übersichtlicher, und hat einen größeren Umfang als das des Behavior. Z.B. kann der Player CD-Laufwerke als Systemkomponente verwalten.

Die Programmierung von .style.time2 ist durch die Verwendung des Players unnötig, setzt aber voraus, dass der Player auch verwendet werden soll.

Es ist zu vermuten, dass die Verwendung des Behavior ressourcenschonender ist, denn es werden nur Eigenschaften und Methoden instanziiert, die für das Behavior-Objekt nötig.

Per .style.time2 Behavior kann der Windows Media Player eingebunden werden, muss aber nicht. Wird der Player eingebunden, so ist er per Behavior ansprechbar. Die Programmierung des Players setzt **aber** eine permanente Player-Instanz im HTML-Dokument voraus und damit ein permanentes Playerfenster. Außerdem wird der Player im **HTML-Dokument** nicht mit allen seine Objekteigenschaften unterstützt, was vielleicht doch wieder für die Nutzung des Behavior .style.time2 spricht.

Behavior .style.time2 wird per XML-Tag in das HTML-Dokument eingebunden. Der Player wird in das HTML-Dokument per Active-X-Control im OBJECT-Tag eingebunden.



Es ist zu vermuten, dass der Behavior ebenfalls das Active-X-Control benutzt, wenn der Windows Media Player in den Behavior eingebunden ist.

Damit wird klar, warum es kein Plugin für den Netscape zum Windows Media Player 7.1 mehr gibt:

Ein Plugin ist kein Active X-Control. Microsoft überlässt es den Fremdanbietern, ihre Playersoftware in Windows 32 Bit per Active-X-Control zu implementieren, da ab IE 6.x generell keine Plugins mehr unterstützt werden. Vorteil ist die Normung per Active-X-Schnittstelle und somit die Komplettintegration des Players und IE in Windows. Nachteil ist das Ausbieten von Konkurrenzprodukten, wenn Microsoft die notwendigen Schnittstellen nicht komplett offengelegt hat und damit eine Weiterentwicklungen durch die Konkurrenz vor allem für Anwender verhindert, die nicht unbedingt mit der Microsoft-Player-Software arbeiten wollen bzw. andere Features erwarten, also sie Microsoft bisher implementiert hat.

4.3.2.2.4.3.39.23.9.1.8. Medien zur Animation per Behavior .style.time2

Media Datei kann lokal auf dem User-PC oder im Netzwerk (z.B. Internet) liegen
Video und Audio in diversen Formen z.B.:

nicht Windows-spezifisch:

- *.AVI
- *.MID
- *.MP3
- *.MPEG
- *.WAV

Windows-spezifisch

- *.WM
- *.WMA
- *.WMV
- *.ASF

<u>Datei-Suffix</u>	<u>MIME type</u>	<u>Windows-Media-Datei</u>
*.wma	audio/x-ms-wma	nur Audio
*.wmv	video/x-ms-wmv	Audio/Video
*.asf	video/x-ms-asf	Audio/Video

hat Media-Typ z.B. "Audio" oder "Video"

kann Attribute folgende Attribute besitzen

"Album"	nur bei Media Item
"Artist"	nicht für Playlist, die per Methode ID_Player.mediaCollection.getByXXX() oder ID_Player.mediaCollection.getAll() erzeugt wurde
"Author"	
"Bitrate"	Bitrate, nur bei Element aus Media-Bibliothek
"Copyright"	nur bei Playlist-Eintrag nicht Playlist von CD nicht Media-Item
"CreationDate"	Datum der Hinzufügen des Elementes zur Media-Bibliothek nicht Playlist und Media Item
"DigitallySecure"	Datum des Schutzes des Elementes in der Media-Bibliothek Schutz = Digital Rights Management nicht Playlist und Media Item
"Genre"	nur Playlist-Eintrag nicht Playlist von CD
"MediaType"	Media-Typ (audio oder video)
"Name"	Name des Playlist-Eintrages
"PlayCount"	Anzahl der Wiederholungen eines Elementes aus der Media-Bibliothek nicht Playlist und Media Item
"SourceURL"	Url oder Pfad und Dateiname der Media-Datei als Element in der Media Bibliothek nicht Playlist und Media Item
"TOC"	CD Table of Contents Identifier nur für Playlist von CD nicht für Playlist, die per Methode ID_Player.mediaCollection.getByXXX() oder ID_Player.mediaCollection.getAll() erzeugt wurde



Die Attribute haben natürlich je nach Media-Datei einen konkreten Inhalt.

Media Clip dasselbe wie Media-Datei, aber nur Video

Meta-Datei Windows-spezifische Script-Datei z.B. ASX-Datei
Die Meta-Elemente in der Datei sind spezifisch zur Art der Meta-Datei.
Jede Meta-Datei kann mindestens 1 Media-Datei und/oder andere Meta-Datei laden:

Bezüglich nicht-Windows-spezifischer Media-Dateien gilt:
Meta-Dateien können sie alle laden.

Bezüglich der Windows-spezifischen Media-Dateien gilt:
Meta-Dateien können nur bestimmte Media-Dateien referenzieren, da
die Kompressionstechnologien nicht in jeder Meta-Datei verfügbar sind.

Übersicht zu Meta-Dateien und ladbaren Windows-spezifischen Media-Dateien

Meta-Datei-Suffix	Windows-Media-Datei-Suffixe		
	*.asf	*.wma	*.wmv
*.wvx	X	X	X
*.wax	X	X	
*.asx	X		

X = ladbar

Übersicht zu Mime-Typen:

Meta-Datei-Suffix	MIME type	Windows-Media-Datei			
*.wax	audio/x-ms-wax	Media-Datei mit Suffix	*.asf	*.wma	
		Meta-Datei mit Suffix		*.wax	
*.wvx	video/x-ms-wvx	Media-Datei mit Suffix		*.wma	*.wmv
		Meta-Datei mit Suffix		*.wax	*.wvx
*.asx	video/x-ms-asf	Media-Datei mit Suffix	*.asf	*.wma	*.wmv
		Meta-Datei mit Suffix	*.asx	*.wax	*.wvx

Playliste ist Liste von mindestens 2 media Objekten aus der Media-Bibliothek
z.B. in Form einer ASX-Meta-Datei
Playliste kann physisch sein lokal oder im Netzwerk
Datenbank oder Textdatei

4.3.2.2.4.3.39.23.9.2. Beschreibung des Behavior

4.3.2.2.4.3.39.23.9.2.1. Syntax

XML <Prefix: CustomTag ID=kette STYLE="behavior:url('#default#time2')" >
HTML <ELEMENT STYLE="behavior:url('#default#time2')" ID=kette>
Scripting object.style.behavior = "url('#default#time2')"
object.addBehavior ("#default#time2")

Hinweis zur XML-Kodierung:
Anstelle der Tagbegrenzung>

sollte ein eigenständiges Endetag kodiert werden, also

.....>
....
</....>

Prefix XML-Namensraum des Behavior
muss immer kodiert werden als
t:xxx

festkodiert sind t und der Doppelpunkt

xxx vordefinierter Bezeichner eines Objektes zum .style.time2 Behavior
z.B.

animation Animationselement
audio Audio-Element
excl Timeline-Container **nur** für Kinder des Elementes
img Image-Element
media Generisches Element
par Neuer Timeline-Container für unabhängiges Element
ref Referenz auf ein generisches Element
seq Neuer Timeline-Container für sequentielle
Zeitsteuerung eines Elementes
video Video-Element
siehe Objektbeschreibungen



Objekte sind die Timer vom Behavior .style.time2. Diese Timer sind vordefiniert und ermöglichen somit nur vordefinierte Arten von Animationen eines Elementes. Die Gestaltung der Animation ist dem Programmierer überlassen.

CustomTag User-definierter Tag
kette String mit dem ID des Elementes (Objektes), das animiert werden soll
muss kodiert werden

Hinweis: Die Kodierung von

```
<ELEMENT STYLE="behavior:url('#default#time2')" ID=kette>
```

ist nicht zu empfehlen, weil unübersichtlich.

**Für alle zu animierenden Elemente und deren Timer sind immer ID-Attribute zu kodieren, auch um die Performance zu erhöhen.
 ID-Attribute sind auch dann nötig, wenn sie nicht zur Referenzierung benutzt werden.
 Empfehlung: Generell im Dokument für jedes Element ein ID kodieren.**

Wenn in den Beispielen der Beschreibungen zum Behavior und seiner Objekte die ID-Attribute nicht immer vollständig kodiert wurden, so dient das **nur** dem Zweck der Übersichtlichkeit:

Es werden in den Beispielen oft nur diejenigen ID kodiert, die auch tatsächlich der Referenzierung dienen.

4.3.2.2.4.3.39.23.9.2.2. Kodierung des .style.time2 Behavior in HTML

```
<HEAD>
<STYLE>
.time_line_klasse { behavior: url('#default#time2') }
</STYLE>
</HEAD>
<BODY>
<html_element_tag ID="ID_des_html_elementes" CLASS="time_line_klasse" ...>
</BODY>
```

time_line_klasse ist ein freier Bezeichner, der als Wert des CLASS-Attributes im zu animierenden Element dient und somit dem Element einen Timer zuordnet

oder

```
<BODY>
....
<html_element_tag ID="ID_des_html_elementes" STYLE="behavior:url('#default#time2')" ...>
....
</BODY>
```

4.3.2.2.4.3.39.23.9.2.3. Import des Behavior .style.time2 in HTML

Der Import des Behavior per HTML wird am einfachsten wie folgt kodiert:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url('#default#time2') }
</STYLE>
</HEAD>
<BODY>
<html_element_tag ID="ID_des_html_elementes" CLASS="time_line_klasse" ...>
</BODY>
```

time_line_klasse ist ein freier Bezeichner, der als Wert des CLASS-Attributes im zu animierenden Element dient und somit dem Element einen Timer zuordnet

oder

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
</HEAD>
<BODY>
....
<html_element_tag ID="ID_des_html_elementes" STYLE="behavior:url('#default#time2')" ...>
....
</BODY>
```

4.3.2.2.4.3.39.23.9.2.4. Bezug des Timers auf ein Element (Objekt) in HTML



4.3.2.2.4.3.39.23.9.2.4.1. *Bezug auf ein Element, das nicht Media-Daten enthält*

Wenn das Element nicht Media-Daten enthält, so muss der Timer auf das Element bezugnehmen, damit er weiss, welches Element er animieren soll. Dazu muss das Element die Timerklasse auch referenzieren.

Beispiel für einen Timer per Objekt animate und einem DIV, der mit dem Timer animiert werden soll

Beispiel 1 für genau 1 Element:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:ANIMATE ID="ID_Animate"
TARGETELEMENT="ID_Div"
ATTRIBUTENAME="left"
TO="400"
DUR="3"
ACCELERATE="1"
REPEATCOUNT="3"
>
</t:ANIMATE>
<DIV ID="ID_Div"
CLASS="time_line_klasse"
STYLE="position: absolute; left:10px.;">
</DIV>
</BODY>
</HTML>
```

Beispiel 2 für Objektgruppe aus verschachtelten Elementen:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2);}
</STYLE>
</HEAD>
<BODY>
<MARQUEE ID="ID_Marquee"
CLASS="time_line_klasse"
TIMECONTAINER="seq"
REPEATCOUNT="indefinite"
>
<IMG ID="ID_Img1" CLASS="time_line_klasse" DUR="4" SRC="test1.gif" ALT="Test1">
<IMG ID="ID_Img2" CLASS="time_line_klasse" DUR="4" SRC="test2.gif" ALT="Test2">
<IMG ID="ID_Img3" CLASS="time_line_klasse" DUR="4" SRC="test3.gif" ALT="Test3">
</MARQUEE>
</BODY>
</HTML>
```

4.3.2.2.4.3.39.23.2.4.2. *Bezug auf ein Element, das nur Media-Daten enthält*

Das Element wird **nur** durch den Timer implementiert.

Die Klassendeklaration innerhalb des HEAD per <STYLE> ... </STYLE> kann daher entfallen.

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.optionale_time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
function Anzeige()
{
alert('hasAudio: ' + ID_Media.hasAudio);
}
</SCRIPT>
</HEAD>
<BODY>
<t:MEDIA ID="ID_Media"
```



```

SRC="test.wmv"
BEGIN="0"
FILL="remove"
onmediacomplete="Anzeige ();"
>
</t:MEDIA>
</BODY>
</HTML>

```

4.3.2.2.4.3.39.23.9.2.4.3. Bezug auf ein Element ohne Media-Daten, das gemeinsam mit Media-Daten animiert werden soll

Wenn das Element zusätzlich zu einer Media-Datei **und mit** der Media-Datei animiert werden soll, dann müssen der Timer das Element und das Element die Timerklasse referenzieren. Dieses Element ist damit synchron zur Media-Datei-Animation. Im Prinzip stellt das eine Gruppe von Elementen dar, denn der Timer muss zwei Objekte animieren.

Beispiel: <HTML XMLNS:t="urn:schemas-microsoft-com:time">
 <HEAD>
 <?IMPORT namespace="t" implementation="#default#time2">
 <STYLE>
 .time_line_klasse { behavior: url(#default#time2) }
 </STYLE>
 </HEAD>
 <BODY>
 <t:VIDEO ID="ID_Video"
 SRC="test.avi"
 SYNCBEHAVIOR="locked"
 >
 </t:VIDEO>
 <SPAN ID="ID_Span"
 CLASS="time_line_klasse"
 DUR="0.1"
 REPEATCOUNT="indefinite"
 onrepeat="ID_Span.innerText= ID_Video.hasDownloadProgress;"
 >

 </BODY>
 </HTML>

4.3.2.2.4.3.39.23.9.2.5. HTML-Style-Attribut im zu animierenden Element

Für die Animation eines Elementes **muss** das STYLE-Attribut im Element kodiert werden. Die Eigenschaften im STYLE-Attribut dienen der Animation per Behavior .style.time2 und seinen Timer-Objekte. .style.time2 kann nur anhand von Style-Eigenschaften animieren. Das trifft auch für filter Objekte zu, die ja ebenfalls im Style kodiert werden.

4.3.2.2.4.3.39.23.9.2.6. Animation eines Elementes anhand einer speziellen Eigenschaft im HTML-Attribut STYLE

Um optional anhand einer speziellen Style-Eigenschaft eines Elementes zu animieren, muss dem Timer die Style-Eigenschaft wie folgt mitgeteilt werden:

```

<t:ANIMATE ID="ID_Animate"
  TARGETELEMENT="ID_Div"
  ATTRIBUTENAME="width"
  ....
>
</t:ANIMATE>

animierter DIV
<DIV ID="ID_Div"
  CLASS="time_line_klasse"
  STYLE= ".....; width=....."
>
</DIV>

```

4.3.2.2.4.3.39.23.9.2.7. Implementierung Timer bzw. Neusetzung seiner Attribute bei aktivem / nichtaktivem Element

Zur Implementierung eines Timers bzw. Neusetzung seiner Attribute darf das Element (Timeline des Elementes) **nicht** bereits aktiv sein. Element beenden per object.endElement();

Nach der Implementierung des Timers bzw. Neusetzung seiner Attribute muss das Element und seine Timeline (neu) gestartet werden. Element starten per object.beginElement();

4.3.2.2.4.3.39.23.9.2.8. Synchronisierung von Timelines

4.3.2.2.4.3.39.23.9.2.8.1. Synchronisierung von verschachtelten Timelines (verschachtelte Timer)

Ziel von verschachtelten Timern ist, dass die Kindertimer das Element des Eltern-Timer **ebenfalls** animieren. Die Kindertimer müssen also zum Eltern-Timer passen. Ideal dafür sind z.B. Filter, die als Kinder ein Bild im Eltern-Timer mitanimieren.

Das Attribut TIMECONTAINER referenziert den Timer-Typ.



Das Attribut **TIMECONTAINER** **muss** in **jedem** Timer, der einen Eltern-Timer hat, kodiert werden, damit der Elterntimer weiss, wie die Kinder das Elternelement mitanimieren. Das ist nötig, um verschachtelte Timer synchronisieren zu können. Der Wert des Attributes im Kind-Timer, also der Typ der Animation per Kind-Timer, muss zur Animation per Eltern-Timer passen. Dafür muss der Programmierer sorgen.

Beispiel: Im Beispiel sind die STYLE-Attributangaben zur Position (z.B. left etc.) nur aus Übersichtlichkeit nicht kodiert worden.

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<DIV ID="ID_Div" STYLE="height:100px">
<t:SEQ ID="ID_Seq"
REPEATCOUNT="indefinite"
>
<t:MEDIA ID="ID_Media1"
SRC = "test1.jpg"
STYLE="position:absolute; ....."
DUR="3"
TIMECONTAINER="par"
FILL="transition"
>
<t:TRANSITIONFILTER ID="ID_Transfilter1"
TYPE="fade"
DUR="2"
>
</t:TRANSITIONFILTER>
<t:TRANSITIONFILTER ID="ID_Transfilter2"
TYPE="ClockWipe"
DUR="2"
>
</t:TRANSITIONFILTER>
</t:MEDIA>
<t:MEDIA ID="ID_Media2"
SRC = "test2.jpg"
STYLE="position:absolute; ....."
DUR="3"
TIMECONTAINER="par"
FILL="transition"
>
<t:TRANSITIONFILTER ID="ID_Transfilter3"
TYPE="ClockWipe"
DUR="2"
>
</t:TRANSITIONFILTER>
<t:TRANSITIONFILTER ID="ID_Transfilter4"
TYPE="fade"
DUR="2"
>
</t:TRANSITIONFILTER>
</t:MEDIA>
</t:SEQ>
</DIV>
</BODY>
</HTML>
```

4.3.2.2.4.3.39.23.9.2.8.2. Zwangs-Synchronisierung von Timelines

Nachfolgendes Beispiel soll die Synchronisierung zeigen zwischen

Timer und genau 1 Element

Timer und Time-Container, wobei der Container eine Elementgruppe einschliesst.

Beispiel für Wiedergabe einer MIDI-Datei per Windows Media Player

MEDIA und ANIMATEMOTION werden per **SYNCBEHAVIOR** zwangssynchronisiert,
wobei der Master der Synchronisation (**SYNCMASTER**) der Timer MEDIA ist.

Ein Master kann **nur** genau einen anderen Timer / Time-Container synchronisieren



(Synchronisationen der Timelines). Der Timer ANIMATEMOTION muss also ohne Attribut **SYNCBEHAVIOR** sein. Um aber trotzdem alle Timer / Time-Container zu synchronisieren, wird das BEGIN-Attribut auf den Wert des gleichnamigen Attributes des Masters gesetzt.

Der Timer-Container t:SEQ wird durch setzen der Eigenschaft BEGIN auf den Wert laut BEGIN vom Timer MEDIA ebenfalls synchronisiert. Damit ist schon deswegen der Start der sequentiellen Animation des DIV1 bis DIV 4 identisch mit dem Start von MEDIA.

Die Animation des DIV5 durch ANIMATEMOTION als Timer vom DIV5 (**TARGETELEMENT="ID_Div5"**) beginnt ebenfalls mit Start von MEDIA: Der Wert des BEGIN-Attributes von ANIMATEMOTION ist identisch mit dem Wert des gleichnamigen Attributes von MEDIA.

Man hätte sich die Kodierung von SYNCMASTER und SYNCBEHAVIOR durch die Wertsetzung der BEGIN-Attribute auf den BEGIN-Wert von MEDIA sparen können.

Die Zwangssynchronisation von MEDIA und dem dem Time-Container SEQ soll hier beispielhaft sein.

Es ist zu beachten, dass Zwangssynchronisation auch Elemente erwartet, die sich ohne Laufzeitprobleme synchronisieren lassen. Eine Synchronisierung kann zur Laufzeit scheitern, was dann ein Rücksetzen der Elemente nachsichziehen muss (ist zu programmieren).

Normalerweise muss zur exakten Behandlung der MIDI-File erst geprüft werden, ob z.B. der Download des MIDI-File erfolgreich erfolgt ist. Erst dann sollte die Midi-Wiedergabe gestartet werden. Um diesen Aufwand zu umgehen, wird in MEDIA der Player als ActiveX-Control verwendet, welcher zugleich Master ist. Damit startet ein zwangssynchronisierter Time-Container auch korrekt mit der Wiedergabe der MIDI-Datei.

Alle Elemente mit Timer-Animation, also alle DIV's, müssen die Timerklasse referenzieren und sind im entsprechenden Timer bzw. Time-Container referenziert.

PLAYER="{6BF52A52-394A-11d3-B153-00C04F79FAA6}" ist das ActiveX-Control für den Windows Media Player 7.1 im Internet Explorer. Ab IE 6.x werden keine Plugins mehr unterstützt.

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY ID="ID_Body">
    <t:PAR ID="ID_Par"
        SYNCBEHAVIOR="locked"
    >
        <t:MEDIA ID="ID_Media"
            SRC="test.mid"
            PLAYER="{6BF52A52-394A-11d3-B153-00C04F79FAA6}"
            SYNCMASTER="true"
            SYNCBEHAVIOR="locked"
            BEGIN="1"
        >
        </t:MEDIA>
        <t:ANIMATEMOTION ID="ID_Animatemotion"
            BEGIN="ID_Media.begin"
            DUR="9"
            TARGETELEMENT="ID_Div5"
            PATH="M 0 0 L 100 300"
            FILL="freeze"
        >
        </t:ANIMATEMOTION>
        <t:SEQ ID="ID_Seq"
            STYLE="font-size:18pt;color:#ff0000"
            SYNCBEHAVIOR="locked"
            BEGIN="ID_Media.begin"
            FILL="freeze"
        >
            <DIV ID="ID_Div1"
                CLASS="time_line_klasse"
                DUR="1.5"
            >
                Diese MIDI-File
            </DIV>
            <DIV ID="ID_Div2"
                CLASS="time_line_klasse"
                DUR="1.5"
            >
```



```

        wird mit dem
    </DIV>
    <DIV ID="ID_Div3"
        CLASS="time_line_klasse"
        DUR="2"
    >
        Windows Media Player
    </DIV>
    <DIV ID="ID_Div4"
        CLASS="time_line_klasse"
        DUR="2"
    >
        abgespielt.
    </DIV>
</t:SEQ>
</t:PAR>
<DIV ID="ID_Div5"
    CLASS="time_line_klasse"
    STYLE="background-color:#FFCC00;position:absolute;height:40;width:120;
        top:70;left:10;font-size:18;border-style:solid
    ">
    >
        animierter DIV waehrend der Wiedergabe der MIDI-File
    </DIV>
</BODY>
</HTML>

```

4.3.2.2.4.3.39.23.9.2.8.3. Synchronisation von Eltern-Time-Container mit seinen Elementen

Die Eigenschaft .endSync beeinflusst das Ende der Animation von Elementen in einem gemeinsamen Time-Container bei Ende der Timeline des Eltern-Time-Containers. Es beschreibt also das Verhalten des Eltern-Time-Containers bezüglich der Animation seiner Kind-Elemente.

Folgende Varianten der Synchronisierung sind möglich:

Eltern-Time-Container endet, wenn alle Kind-Elemente geendet sind.

Wenn Kinder mit endloser Wiederholung: Alle endlos zu wiederholende Kinder werden so lange wiederholt, bis die anderen Kinder ohne Endlos-Animation enden. Danach endet der Eltern-Timer-Container und alle Endlos-wiederholungen werden zugleich gestoppt.

Eltern-Container endet, wenn genau ein Kind-Element geendet ist (egal welches und wann gestartet).

Wenn Kinder mit endloser Wiederholung: Endlos-Wiederholungen eines Kindes können nicht zum Ende des Eltern-Time-Container führen. Der Container muss mindestens 1 Element ohne Endloswiederholung besitzen.

Eltern-Container endet, wenn das Element laut ID-Attribut endet. Dieses Element darf natürlich keine Endloswiederholung besitzen.

Standardgemäß endet der Eltern-Time-Container, wenn alle Kind-Elemente geendet sind.

Wenn Kinder mit endloser Wiederholung: Endlos-Wiederholungen eines Kindes können nicht zum Ende des Eltern-Time-Container führen.

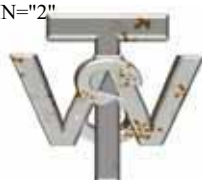
Eltern-Time-Container wartet nie auf das Ende der Kinder, sondern endet immer mit Ende der Timeline des Eltern-Time-Containers. Wenn Kinder mit endloser Wiederholung: Endlos-Wiederholungen eines Kindes werden mit Ende des Eltern-Time-Container sofort gestoppt.

Beispiel:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
    <t:EXCL ID="ID_Excl"
        ENDSYNC="ID_Div2"
    >
        <DIV ID="ID_Div1"
            CLASS="time_line_klasse"
            BEGIN="0"
            DUR="2"
        >
            Zeile 1
        </DIV>
        <DIV ID="ID_Div2"
            CLASS="time_line_klasse"
            BEGIN="2"

```



```

        DUR="2"
    >
        Zeile 2
    </DIV>
    <DIV    ID="ID_Div3"
        CLASS="time_line_klasse"
        BEGIN="3"
        DUR="2"
    >
        Zeile 3
    </DIV>
</t:EXCL>
</BODY>
</HTML>

```

4.3.2.2.4.3.39.23.9.2.9. Beispiele in HTML

Beispiel 1: pixelweise Ausdehnung eines DIV in seiner Breite (Style-Eigenschaft .width) nach rechts

Es wurden für jedes Element im Dokument ein ID kodiert, also auch dann, wenn das ID keiner Referenzierung dient.

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
    function SpanInhaltInit(Kette)
    {
        // Zustand
        ID_Span1.innerText = "";

        // Kumulationsart
        ID_Span2.innerText =Kette;

        // Zaehler auf 1
        ID_Span3.innerText ="1";

        // DIV-Text festlegen
        ID_Div.innerText="Dieser DIV dehnt sich in der Breite ";
        if (Kette == "sum")
        {
            // Wertkumulation von .width
            ID_Div.innerText += " genau 1x kumulativ um ";
            ID_Div.innerText += ID_Animate.by
            ID_Div.innerText += " mal "
            ID_Div.innerText += ID_Animate.repeatCount;
        }
        else
        {
            // keine Wertkumulation von .width
            ID_Div.innerText += ID_Animate.repeatCount;
            ID_Div.innerText += " mal um ";
            ID_Div.innerText += ID_Animate.by
        }

        ID_Div.innerText += " aus";
    }

    function KumulationAus()
    {
        // Element darf nicht aktiv sein !!

        // ERST .accumulate definieren
        SpanInhaltInit("none");
        ID_Animate.accumulate="none";

        // DANACH Animation starten
        ID_Animate.beginElement();
    }

    function KumulationEin()
    {

```



```

        // Element darf nicht aktiv sein !!

        // ERST .accumulate definieren
        SpanInhaltInit("sum");
        ID_Animate.accumulate="sum";

        // DANACH Animation starten
        ID_Animate.beginElement();
    }

    function Anzeige()
    {
        ID_Span1.innerHTML = "Animation ist beendet ";
    }
}
</SCRIPT>
</HEAD>
<BODY>
    <t:ANIMATE
        ID="ID_Animate"
        TARGETELEMENT="ID_Div"
        ATTRIBUTENAME="width"
        BY="150px"
        DUR="3"
        REPEATCOUNT="3"
        BEGIN="indefinite"
        FILL="freeze"
        onend="Anzeige();"
        onrepeat="ID_Span3.innerHTML= ID_Animate.currTimeState.repeatCount + 1;"
    >
    <t:ANIMATE>

    animierter DIV
    <DIV
        ID="ID_Div"
        CLASS="time_line_klasse"
        STYLE= "position:absolute;
            top:125px;
            left:25px;
            height:100px;
            width:125px;
            border:solid black 2px;
            "
    >
    </DIV>
    <BR>

    Zustand der Animation:
    <SPAN ID="ID_Span1"></SPAN>
    <BR>

    Kumulationsart:
    <SPAN ID="ID_Span2"></SPAN>
    <BR>

    Durchlaufzaehler:
    <SPAN ID="ID_Span3"></SPAN>
    <BR>

    <BUTTON ID="ID_Button1" onclick="KumulationAus()">Kumulation aus </BUTTON>
    <BUTTON ID="ID_Button2" onclick="KumulationEin()">Kumulation ein</BUTTON>
</BODY>
</HTML>

```

Beispiel 2 Beispiel für Wiedergabe per Windows Media Player **und** Synchronisation von Elementgruppen:

MEDIA und ANIMATEMOTION werden per **SYNCBEHAVIOR** zwangssynchronisiert, wobei der Master der Synchronisation (**SYNCMASTER**) der Timer MEDIA ist.

Ein Master kann **nur** genau einen anderen Timer / Time-Container synchronisieren (Synchronisationen der Timelines). Der Timer ANIMATEMOTION muss also ohne Attribut **SYNCBEHAVIOR** sein. Um aber trotzdem alle Timer / Time-Container zu synchronisieren, wird das BEGIN-Attribut auf den Wert des gleichnamigen Attributes des Masters gesetzt.

Der Timer-Container t:SEQ wird durch setzen der Eigenschaft BEGIN auf den Wert laut BEGIN vom Timer MEDIA ebenfalls synchronisiert. Damit ist schon deswegen der Start der sequentiellen Animation des DIV1 bis DIV 4 identisch mit dem Start von MEDIA.

Die Animation des DIV5 durch ANIMATEMOTION als Timer vom DIV5



(**TARGETELEMENT**="ID_Div5") beginnt ebenfalls mit Start von MEDIA:

Der Wert des BEGIN-Attributes von ANIMATEMOTION ist identisch mit dem Wert des gleichnamigen Attributes von MEDIA.

Man hätte sich die Kodierung von SYNCMASTER und SYNCBEHAVIOR durch die Wertsetzung der BEGIN-Attribute auf den BEGIN-Wert von MEDIA sparen können.

Die Zwangssynchronisation von MEDIA und dem dem Time-Container SEQ soll hier beispielhaft sein.

Es ist zu beachten, dass Zwangssynchronisation auch Elemente erwartet, die sich ohne Laufzeitprobleme synchronisieren lassen. Eine Synchronisierung kann zur Laufzeit scheitern, was dann ein Rücksetzen der Elemente nachsichziehen muss (ist zu programmieren).

Normalerweise muss zur exakten Behandlung der MIDI-File erst geprüft werden, ob z.B. der Download des MIDI-File erfolgreich erfolgt ist. Erst dann sollte die Midi-Wiedergabe gestartet werden. Um diesen Aufwand zu umgehen, wird in MEDIA der Player als ActiveX-Control verwendet, welcher zugleich Master ist. Damit startet ein zwangssynchronisierter Time-Container auch korrekt mit der Wiedergabe der MIDI-Datei.

Alle Elemente mit Timer-Animation, also alle DIV's, müssen die Timerklasse referenzieren und sind im entsprechenden Timer bzw. Time-Container referenziert.

PLAYER="{6BF52A52-394A-11d3-B153-00C04F79FAA6}" ist das ActiveX-Control für den Windows Media Player 7.1 im Internet Explorer. Ab IE 6.x werden keine Plugins mehr unterstützt.

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY ID="ID_Body">
  <t:PAR ID="ID_Par"
    SYNCBEHAVIOR="locked"
  >
    <t:MEDIA ID="ID_Media"
      SRC="test.mid"
      PLAYER="{6BF52A52-394A-11d3-B153-00C04F79FAA6}"
      SYNCMASTER="true"
      SYNCBEHAVIOR="locked"
      BEGIN="1"
    >
    </t:MEDIA>
    <t:ANIMATEMOTION ID="ID_Animatemotion"
      BEGIN="ID_Media.begin"
      DUR="9"
      TARGETELEMENT="ID_Div5"
      PATH="M 0 0 L 100 300"
      FILL="freeze"
    >
    </t:ANIMATEMOTION>
    <t:SEQ ID="ID_Seq"
      STYLE="font-size:18pt;color:#ff0000"
      SYNCBEHAVIOR="locked"
      BEGIN="ID_Media.begin"
      FILL="freeze"
    >
      <DIV ID="ID_Div1"
        CLASS="time_line_klasse"
        DUR="1.5"
      >
        Diese MIDI-File
      </DIV>
      <DIV ID="ID_Div2"
        CLASS="time_line_klasse"
        DUR="1.5"
      >
        wird mit dem
      </DIV>
      <DIV ID="ID_Div3"
        CLASS="time_line_klasse"
        DUR="2"
      >
        Windows Media Player
```



```

        </DIV>
        <DIV ID="ID_Div4"
            CLASS="time_line_klasse"
            DUR="2"
        >
            abgespielt.
        </DIV>
    </t:SEQ>
</t:PAR>
<DIV ID="ID_Div5"
    CLASS="time_line_klasse"
    STYLE="background-color:#FFCC00;position:absolute;height:40;width:120;
        top:70;left:10;font-size:18;border-style:solid
    "
>
    animierter DIV waehrend der Wiedergabe der MIDI-File
</DIV>
</BODY>
</HTML>

```

4.3.2.2.4.3.39.23.9.2.9.

Eigenschaften

.accelerate Beschleunigung des Elementes auf der Timeline
hat keinen Einfluss auf die Dauer der Timeline
auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur
Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten
wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
siehe .decelerate

.autoReverse automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation
des Elementes auf der Timeline

.begin Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction

.decelerate Verlangsamung des Elementes auf der Timeline
hat keinen Einfluss auf die Dauer der Timeline
auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur
Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten
wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
siehe .accelerate

.dur Dauer Objektaktivitäten laut Eigenschaft .timeAction
alternativ: Eigenschaft .end

.end Objektaktivitäten laut Eigenschaft .timeAction beenden
ab IE 6.x
alternativ: Eigenschaft .dur

.fill Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber
bevor die Timeline des Elternelementes endet
ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet
werden darf !!

.hasMedia Objekt ist HTML-Media-Objekt

.mute Audio aktiv oder aus (stumm) auf der Timeline
wenn Eigenschaft für body Objekt kodiert, so alle Audio-Elemente im Body davon betroffen
beeinflusst nicht die Systemlautstärke-Regelung von Windows oder anderen Soundmixern
siehe Objekt bgsound

.repeatDur Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline
verlangt kodierte Eigenschaft .dur oder .repeat
darf nicht kodiert werden mit der Eigenschaft .repeatCount

.restart generelle Restartmöglichkeit eines Elementes auf der Timeline
ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!

.speed aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)

.syncBehavior Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes
sinnvoll vorallem bei Time-Container (Gruppen von Objekten)
Es muss die Eigenschaft .syncmaster kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der
Beschreibung zum jeweiligen Objekt zu entnehmen.
siehe .syncTolerance und .syncmaster

.systemBitrate wird hier nicht erklärt

.systemCaptions wird hier nicht erklärt

.systemLanguage Sprache festlegen für das Objekt

.systemOverdubOrSubtitle wird hier nicht erklärt

.timeAction Aktion des Objektes in der Timeline
Aktion nur ausführbar wenn Objekt aktiv ist UND Timeline aktiv ist

.timeContainer Typ der Timeline des Objektes

.timeParent Zeiger auf das Eltern-Timeline

.volume Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes
für alle Audio-Elemente des body Objektes auch möglich
Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus !
siehe Objekt bgsound

4.3.2.2.4.3.39.23.9.2.10.

Methoden



<code>.activeTimeToParentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.activeTimeToSegmentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.beginElement()</code>	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes.. per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.beginElementAt()</code>	Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.documentTimeToParentTime()</code>	Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
<code>.endElement()</code>	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes.. per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.endElementAt()</code>	aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.parentTimeToActiveTime()</code>	Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.parentTimeToDocumentTime()</code>	Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
<code>.pauseElement()</code>	aktives Element auf Timeline pausieren lassen ersetzt Methode <code>pause()</code> , die deprecated ist und nicht mehr verwendet werden darf ! per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.resetElement()</code>	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
<code>.resumeElement()</code>	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben falls Element nicht pausiert, passiert nichts ersetzt die Methode <code>resume()</code> , da sie deprecated ist und nicht mehr verwendet werden darf per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.seekActiveTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Timeline wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen alle Media-Typen für Element zulässig siehe Eigenschaft <code>.canSeek</code>
<code>.seekSegmentTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline ohne Wiederholung der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.seekTo()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.seekToFrame()</code>	Frame eines aktiven Elementes auf der Timeline anwählen wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.segmentTimeToActiveTime()</code>	Wert der Segment-Timeline des Elements in den korrespondierenden Wert



	der aktiven Timeline des Elements konvertieren
.segmentTimeToSimpleTime()	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
.simpleTimeToSegmentTime()	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren
4.3.2.2.4.3.39.23.9.2.11. Events	
onbegin	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Element erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Eltern-Element siehe onend und onrepeat
onend	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut .repeatCount bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit des Kindelement ebenfalls endet nicht erzeugt wenn .fill mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode .endElement() und Eigenschaft .end siehe onbegin und onrepeat
onmediacomplete	erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde für Animation per Timeline Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline siehe onmediaerror
onmediaerror	erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält für Animation per Timeline Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline ersetzt das Ereignis onmedialoadfailed, das deprecated ist und nicht mehr verwendet werden darf ! siehe onmediacomplete
onpause	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt
onrepeat	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe onbegin), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für onrepeat kodiert hat (kein Heraufreichen des Ereignisses onrepeat zum Elternelement) .repeatCount bzw. .repeat muss > 1 sein: onrepeat wird also .repeatCount - 1 mal erzeugt Kind eines Elementes siehe onend, onbegin, .repeatCount und .repeat
onreset	erzeugt wenn Element zurückgesetzt wurde per Methode .resetElement(), also die Timeline des Elementes den aktuellen Wert laut .begin wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
onresume	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
onreverse	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist) .autoReverse muss auf "true" stehen
onseek	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde: .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame()

4.3.2.2.4.3.39.23.9.2.12. Objekte und Collectionen zur Verwaltung von style.time2

4.3.2.2.4.3.39.23.9.2.12.1. currTimeState Objekt des Internet Explorer

verwaltet die **aktuelle** HTML- und Zeitsteuerung (Timeline) der Animation eines Elementes und seiner Kinder
Elemente sind z.B. Sound, Videos aber auch Nicht-Media-Objekte

komplett erst ab IE 6.x unterstützt

nur in Verbindung mit dem Objekten time2 (siehe dort) z.B. transitionFilter

Die Kodierung der Eigenschaften muss erfolgen per id_des_objektes.currTimeState.eigenschaft

Hinweis: Objekt time ist deprecated und durch time2 ersetzt worden.



Beispiel 1:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:PAR SPEED="2" >
    <t:KIND ID="ID_Kind"
        BEGIN="1"
        SRC="http://www.test.de/media/test.wmv"
        SPEED="1.5"
    >
    </t:KIND>
</t:PAR>
<BUTTON ID="ID_Button1"
    CLASS="time_line_klasse"
    REPEATCOUNT="indefinite"
    onclick="alert( 'Run time speed des Kindes: ' + ID_Kind.currTimeState.speed );"
>
    Run time speed des Kindes
</BUTTON>
<BUTTON ID="ID_Button2"
    CALSS="time_line_klasse"
    REPEATCOUNT="indefinite"
    onclick="alert( 'SPEED-Attribut des Kindes: ' + ID_Kind.speed );"
>
    SPEED-Attribut des Kindes
</BUTTON>
</BODY>
</HTML>

```

Beispiel 2:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<SCRIPT>
function Aendern()
{
    ID_Video.updateMode = ID_Select1.options.value;
    ID_Video.speed      = ID_Select1.options.value;
}
</SCRIPT>
</HEAD>
<BODY>
<t:VIDEO ID="ID_Video"
    SRC="test.avi"
    UPDATEMODE="reset"
    STYLE="width:175px; height:150px;"
>
</t:VIDEO>
<BR>
updateMode waehlen:
<SELECT NAME="ID_Select1">
    <OPTION VALUE="auto">Auto</OPTION>
    <OPTION VALUE="reset" SELECTED>Reset</OPTION>
</SELECT>
<BR>
Geschwindigkeit waehlen:
<SELECT NAME="ID_Select2">
    <OPTION VALUE="0.25">25%</OPTION>
    <OPTION VALUE="0.50">50%</OPTION>
    <OPTION VALUE="0.75">75%</OPTION>
    <OPTION VALUE="1" SELECTED>100% </OPTION>
    <OPTION VALUE="2" SELECTED>200% </OPTION>
</SELECT>
<BR>

```



```
<BUTTON ID="ID_Button1" onClick="Aendern();">
    Geschwindigkeit aendern
</BUTTON>
<BUTTON ID="ID_Button1" onClick="document.body.beginElement(">
    Restart
</BUTTON>
</CENTER>
</BODY>
</HTML>
```

Beispiel 3:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
function Stummschaltung()
{
    if (ID_Video.currTimeState.isActive)
    { ID_Span.innerText=' stumm = ' + ID_Video.currTimeState.isMuted;}
    else
    { ID_Span.innerText=' stumm =';}
}
</SCRIPT>
</HEAD>
<BODY>
<t:media ID="ID_Video"
    SRC="/test /media/test.wmv"
    BEGIN="indefinite;"
    FILL="remove"
>
</ t:media>
<BR><BR>
<SPAN ID="ID_Span">stumm =</SPAN>
<BR><BR>
<BUTTON onclick="ID_Video.beginElement();Stummschaltung();">
    Start
</BUTTON>
<BUTTON onclick="ID_Video.endElement();Stummschaltung();">
    Stop
</BUTTON>
<BR><BR>
<B>Volume control:</B>&nbsp;
<INPUT TYPE="radio"
    onpropertychange=
        "ID_Video.mute='true'; Stummschaltung();"
>
    Mute
<INPUT TYPE="radio" CHECKED
    onpropertychange= "ID_Video.mute='false'; ID_Video.volume=100; Stummschaltung();"
>
    100% Volume
</BODY>
</HTML>
```

Eigenschaften:

<code>.activeDur</code>	totale Dauer der Timeline in Sekunden inklusive aller Wiederholungen und Autoreverse
<code>.activeTime</code>	aktueller Zeitpunkt in Sekunden in der Timeline ab Start inklusive aller Repeats, autoReverse
<code>.isActive</code>	Elemente-Status der Aktivität eines Elementes in der Timeline Hinweis: Status der Timeline per Eigenschaft <code>.state</code> oder <code>.stateString</code> abfragen
<code>.isMuted</code>	Elemente-Status der Audio-Stummschaltung in der Timeline
<code>.isOn</code>	Elemente-Status aktiv oder halten in der Timeline Staats aktiv: Objekt kann auf Events reagieren Status halten: Attribut <code>Fill</code> muss auf "freeze" oder "hold" gesetzt sein ermittelbar per eigenschaft <code>.state</code> Objekt kann nicht auf Events reagieren zu Attribut <code>FILL</code> : wenn "hold" oder freeze" dann wartet Element auf die Synchronisation mit der Timeline des Elternobjektes wenn "hold" Element kann keine Events verarbeiten



ist inaktiv

	wenn "freeze" Element ist aktiv
.isPaused	Elemente-Status pausieren in der Timeline
.parentTimeBegin	Startzeit als Offset von der Startzeit der Eltern-Timeline
.parentTimeEnd	Endezeit als Offset von der Startzeit der Eltern-Timeline
	ist zugleich Summe der Werte laut Eigenschaften parentTimeBegin und activeDur
.progress	Hinweis zu activeDur: inklusive aller Wiederholungen und Autoreverse Fortschrittangabe entlang der Timeline inklusive Repeats etc.
.repeatCount	aktuelle Nummer der Wiederholung bei Loop
.segmentDur	Dauer der Timeline der Wiederholungen laut autoReverse
.segmentTime	aktueller Zeitpunkt in der Timeline der Wiederholungen laut autoReverse
.simpleDur	Dauer einer Wiederholung
.simpleTime	Wiederholung erzeugt per Eigenschaft .autoReverse auf true setzen aktueller Punkt auf Timeline
.speed	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed) Beispiel: Eltern mit 50 % Geschwindigkeit Kind mit 50 % Geschwindigkeit der Eltern also aktuelle Wiedergabegeschwindigkeit = 25 %
.state	Status der Timeline – Variante 1
.stateString	Status der Timeline – Variante 2
.volume	Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes für alle Audio-Elemente des body Objektes auch möglich Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus ! siehe Objekt currTimeState und Behavior .style.time2 siehe Objekt bgsound

Methoden:

keine

4.3.2.2.4.3.39.23.9.2.12.2. timeChildren Collection des Internet Explorer

Feld aller getimter Kinder eines Elementes

Syntax:

```
[ var FeldZeiger = ] object.timeChildren;

[ var FeldElementZeiger = ] object.timeChildren[Index];
```

Index: Integer und ab 0
muss in [] kodiert sein

Eigenschaften:

.length Anzahl der Feldelemente also Feldlänge z.B. bei Collection

Methoden:

.item() Referenz auf Feldelement anhand des Integer-Indexes oder des
Attributnamen (analog zu ID oder NAME-Attribut) liefern
außer bei Formular mit <INPUT TYPE=image ...>
da dafür die children-Collection verwendet werden muss !!!

4.3.2.2.4.3.39.23.9.3. Behavior-Objekte und Behavior-Collections

Nachfolgende Objekte sind die Timer vom Behavior .style.time2. Diese Timer sind vordefiniert und ermöglichen somit nur vordefinierte Arten von Animationen eines Elementes. Die Gestaltung der Animation ist dem Programmierer überlassen.

Als timerorientiertes Objekt ist die Playliste anhand der Objektes playItem und der Behavior-Collection .style.time2.playList zu nennen.

Zur Implementierung eines Timers bzw. Neusetzung seiner Attribute darf das Element (Timeline des Elementes) nicht bereits aktiv sein.
Element beenden per object.endElement();

Nach der Implementierung des Timers bzw. Neusetzung seiner Attribute muss das Element und seine Timeline (neu) gestartet werden.
Element starten per object.beginElement();

4.3.2.2.4.3.39.23.9.3.1. .style.time2.animate Behavior-Objekt des Internet Explorer

Timerobjekt, das zur Animation eines Elementes anhand einer speziellen Style-Eigenschaft des Elementes dient. Die Style-Eigenschaft muss im STYLE-Attribut des Elementes **und** im ATTRIBUTENAME-Attribut des Timers kodiert sein.

Syntax:

XML t:ANIMATE
Script animate

Beispiel: pixelweise Ausdehnung eines DIV in seiner Breite (Style-Eigenschaft .width) nach rechts:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
```



```

function SpanInhaltInit(Kette)
{
    // Zustand
    ID_Span1.innerText = "";

    // Kumulationsart
    ID_Span2.innerText =Kette;

    // Zaehler auf 1
    ID_Span3.innerText ="1";

    // DIV-Text festlegen
    ID_Div.innerText="Dieser DIV dehnt sich in der Breite ";
    if (Kette == "sum")
    {
        // Wertkumulation von .width
        ID_Div.innerText += " genau 1x kumulativ um ";
        ID_Div.innerText += ID_Animate.by
        ID_Div.innerText += " mal "
        ID_Div.innerText += ID_Animate.repeatCount;
    }
    else
    {
        // keine Wertkumulation von .width
        ID_Div.innerText += ID_Animate.repeatCount;
        ID_Div.innerText += " mal um ";
        ID_Div.innerText += ID_Animate.by
    }

    ID_Div.innerText += " aus";
}

function KumulationAus()
{
    // Element darf nicht aktiv sein !!

    // ERST .accumulate definieren
    SpanInhaltInit("none");
    ID_Animate.accumulate="none";

    // DANACH Animation starten
    ID_Animate.beginElement();
}

function KumulationEin()
{
    // Element darf nicht aktiv sein !!

    // ERST .accumulate definieren
    SpanInhaltInit("sum");
    ID_Animate.accumulate="sum";

    // DANACH Animation starten
    ID_Animate.beginElement();
}

function Anzeige()
{
    ID_Span1.innerText = "Animation ist beendet ";
}
</SCRIPT>
</HEAD>
<BODY>
    <t:ANIMATE
        ID="ID_Animate"
        TARGETELEMENT="ID_Div"
        ATTRIBUTENAME="width"
        BY="150px"
        DUR="3"
        REPEATCOUNT="3"
        BEGIN="indefinite"
        FILL="freeze"
        onend="Anzeige();"
        onrepeat="ID_Span3.innerText= ID_Animate.currTimeState.repeatCount + 1;"

```



```

>
<t:ANIMATE>

animierter DIV
<DIV ID="ID_Div"
  CLASS="time_line_klasse"
  STYLE="position:absolute;
        top:125px;
        left:25px;
        height:100px;
        width:125px;
        border:solid black 2px;
        "
>
</DIV>
<BR>

Zustand der Animation:
<SPAN ID="ID_Span1"></SPAN>
<BR>

Kumulationsart:
<SPAN ID="ID_Span2"></SPAN>
<BR>

Durchlaufzaehler:
<SPAN ID="ID_Span3"></SPAN>
<BR>

<BUTTON ID="ID_Button1" onclick="KumulationAus()">Kumulation aus </BUTTON>
<BUTTON ID="ID_Button2" onclick="KumulationEin()">Kumulation ein</BUTTON>
</BODY>
</HTML>

```

Eigenschaften:

.accelerate
Beschleunigung des Elementes auf der Timeline
hat keinen Einfluss auf die Dauer der Timeline
auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur
Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten
wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
siehe .decelerate

.accumulate
Kumulative Animation eines Elementes auf der Timeline
Element darf nicht aktiv sein, wenn .accumulate definiert werden soll:
Element erst danach starten.
Es kann jede numerische Style-Eigenschaft kumulativ verändert werden und damit
die kumulative Animation des Elementes in der Style-Eigenschaft erzeugen.
Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des
Wertes sein (mathematische Berechnungen erst nach Entfernung der Einheit
per Stringoperationen möglich)
Name der Eigenschaft laut .attributName
Änderung des Wertes pro Durchlauf bzw. kumulativ um den Maximalwert laut .by
Anzahl der Durchläufe bzw. der Wert-Kumulationen laut .repeatCount
Kumulationsschrittweite laut .calcMode
Ereignis des Startes eines Durchlaufes bzw. Kumulation um Maximalwert ist onrepeat.
Wert der Eigenschaft .repeatCount muss > 1 sein
Kumulation nicht aktiv:
Nach jedem Durchlauf laut .repeatCount erfolgt Rücksetzen der Style-Eigenschaft des
Elementes auf den Zustand vor dem Durchlaufbeginn.
Es wird also die Style-Eigenschaft nicht wertmäßig kumuliert.
Das Element wird also laut .repeatCount mehrmals animiert.
Pro Durchlauf wird für die Eigenschaft laut .attributName der Wert
ab Startwert
in Schrittweite laut .calcMode
um den maximalen Wert laut .by
erhöht.
Kumulation aktiv:
Die Style-Eigenschaft des Elementes wird pro Durchlauf wertmäßig laut
.repeatCount um den Wert laut .by kumuliert.
Kumulationsschrittweite laut .calcMode
Das Element wird also genau 1 mal animiert.
Mit Beginn des Startes des Elementes wird für die Eigenschaft laut .attributName
der Wert ab Startwert
in Schrittweite laut .calcMode
um den maximalen Wert aus dem Produkt aus
.by mal Anzahl der Wiederholungen



z.B. laut .repeatCount

erhöht.

Es wird also über **alle** Wiederholungen der Wert kumuliert.

	siehe .additive .calcMode
.additive	numerische Werte von Eigenschaften eines Elementes ersetzen oder zu den numerischen Werten der gleichnamigen Eigenschaften anderer Elemente addieren während der Elemente-Animation auf der Timeline Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des Wertes sein gültige Eigenschaften sind z.B. .values oder .by sinnvoll für Kombinationen von Elementen (auch bei Wiederholung eines Elementes) Element darf nicht aktiv sein, wenn .additive definiert werden soll: Element erst danach starten. Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert !
.attributeName	siehe .accumulate .by .attributeName Bezeichner einer Style-Eigenschaft, also Name eines Attributes, für die Animation anhand dieses Attributes auf der Timeline
.autoReverse	siehe .accumulate .by .additive .calcMode automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.begin	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction
.by	Wert um den die Werterhöhung bei Elemente-Animation(en) auf der Timeline per .additive oder .accumulate erfolgen soll Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des Wertes sein wird für Animation in Einheiten zerlegt laut Schrittweite laut .calcMode für die Objekte animate, animateMotion und animateColor gilt: .by wird von .path, .to und .values überschrieben Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert ! siehe .attributeName .accumulate .additive .calcMode
.calcMode	Schrittweite der Werterhöhung bzw. Art der Animation in der Ebene (1D oder 2D) bei Elemente-Animation(en) auf der Timeline per .additive oder .accumulate
.decelerate	Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert ! Verlangsamung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .accelerate
.dur	Dauer Objektaktivitäten laut Eigenschaft .timeAction
.end	alternativ: Eigenschaft .end Objektaktivitäten laut Eigenschaft .timeAction beenden ab IE 6.x alternativ: Eigenschaft .dur
.fill	Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber bevor die Timeline des Elternelementes endet ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet werden darf !!
.from	Startwert zur Werterhöhung bei Elemente-Animation(en) auf der Timeline per .additive oder .accumulate für die Objekte animate, animateMotion und animateColor gilt: .from wird von .path und .values überschrieben Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert !
.hasMedia	Objekt ist HTML-Media-Objekt
.keySplines	Wert eines Pixel-Intervalles zur Werterhöhung bei 2D-Elemente-Animation(en) auf der Timeline per .additive oder .accumulate benötigt .calcMode auf "spline" .keyTimes .values oder .path
.keyTimes	Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert ! Zeitwert eines Pixel-Intervalles zur Werterhöhung bei 2D-Elemente-Animation(en) auf der Timeline per .additive oder .accumulate benötigt .calcMode auf "spline" .keySplines .values oder .path
.repeatCount	Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert ! aktuelle Nummer der Wiederholung bei Loop
.repeatDur	Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline verlangt kodierte Eigenschaft .dur oder .repeat darf nicht kodiert werden mit der Eigenschaft .repeatCount
.restart	generelle Restartmöglichkeit eines Elementes auf der Timeline ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!
.speed	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)



<code>.systemBitrate</code>	wird hier nicht erklärt
<code>.systemCaptions</code>	wird hier nicht erklärt
<code>.systemLanguage</code>	Sprache festlegen für das Objekt
<code>.systemOverdubOrSubtitle</code>	wird hier nicht erklärt
<code>.targetElement</code>	ID des zu animierenden Elementes auf der Timeline muss für Kindelement immer kodiert werden, wenn nicht das Elternelement animiert werden soll muss nicht kodiert werden, wenn kein Elternelement vorliegt Achtung: Objekt body ist das oberste Elternelement im Dokument und somit haben Elemente innerhalb BODY immer Eltern Empfehlung: immer kodieren ist der Bezug des Time-Container auf das zu animierende Element
<code>.timeContainer</code>	Typ der Timeline des Objektes
<code>.timeParent</code>	Zeiger auf das Eltern-Timeline
<code>.to</code>	Endwert zur Werterhöhung bei Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> für die Objekte <code>animate</code> , <code>animateMotion</code> und <code>animateColor</code> gilt: <code>.to</code> wird von <code>.path</code> und <code>.values</code> überschrieben <code>.to</code> überschreibt <code>.by</code>
<code>.values</code>	Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert ! 2D-Animation bei der Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> anhand absoluter Koordinaten im Grafiksystem siehe <code>.path</code> für Vektorgraphik-Animation anhand relativer und absoluter Koordinaten Werteliste korrespondiert zu der Werteliste der Eigenschaft <code>.keyTimes</code> benötigt <code>.calcMode</code> auf "spline" <code>.keySplines</code> und <code>.keyTimes</code> für die Objekte <code>animate</code> , <code>animateMotion</code> und <code>animateColor</code> gilt: <code>.values</code> wird von <code>.path</code> überschrieben <code>.values</code> überschreibt <code>.by</code> <code>.from</code> <code>.to</code> Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert !

Methoden:

<code>.activeTimeToParentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.activeTimeToSegmentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.beginElement()</code>	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
<code>.beginElementAt()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start
<code>.documentTimeToParentTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
<code>.endElement()</code>	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
<code>.endElementAt()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne
<code>.parentTimeToActiveTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren
<code>.parentTimeToDocumentTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
<code>.pauseElement()</code>	aktives Element auf Timeline pausieren lassen ersetzt Methode <code>pause()</code> , die deprecated ist und nicht mehr verwendet werden darf ! per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.resetElement()</code>	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
<code>.resumeElement()</code>	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben



	<p>falls Element nicht pausiert, passiert nichts</p> <p>ersetzt die Methode <code>resume()</code>, da sie deprecated ist und nicht mehr verwendet werden darf</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p>
<code>.seekActiveTime()</code>	<p>aktives Element animieren ab einem Zeitpunkt auf der Timeline</p> <p>wenn Element nicht aktiv, so Fehler erzeugt</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p>
<code>.seekSegmentTime()</code>	<p>alle Media-Typen für Element zulässig</p> <p>siehe Eigenschaft <code>.canSeek</code></p> <p>aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline</p> <p>ohne Wiederholung der Animation</p> <p>wenn Element nicht aktiv, so Fehler erzeugt</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p> <p>Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst</p> <p>nicht alle Media-Typen für Element zulässig</p> <p>wenn unzulässig, so kein Fehler</p>
<code>.seekTo()</code>	<p>siehe Eigenschaft <code>.canSeek</code></p> <p>aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline</p> <p>einschliesslich möglicher Wiederholungen der Animation</p> <p>wenn Element nicht aktiv, so Fehler erzeugt</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p> <p>nicht alle Media-Typen für Element zulässig</p> <p>wenn unzulässig, so kein Fehler</p>
<code>.seekToFrame()</code>	<p>siehe Eigenschaft <code>.canSeek</code></p> <p>Frame eines aktiven Elementes auf der Timeline anwählen</p> <p>wenn Element nicht aktiv, so Fehler erzeugt</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p> <p>nicht alle Media-Typen für Element zulässig</p> <p>wenn unzulässig, so kein Fehler</p>
<code>.segmentTimeToActiveTime()</code>	<p>siehe Eigenschaft <code>.canSeek</code></p> <p>Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren</p>
<code>.segmentTimeToSimpleTime()</code>	<p>Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren</p>
<code>.simpleTimeToSegmentTime()</code>	<p>Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren</p>
Events:	
<code>onbegin</code>	<p>erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird</p> <p>nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Element</p> <p>erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Eltern-Element</p> <p>siehe <code>onend</code> und <code>onrepeat</code></p>
<code>onend</code>	<p>erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut <code>.repeatCount</code></p> <p>bzw. wenn das aktive Element gestoppt wird</p> <p>erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit das Kindelement ebenfalls endet</p> <p>nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht</p> <p>siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code></p> <p>siehe <code>onbegin</code> und <code>onrepeat</code></p>
<code>onpause</code>	<p>erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren</p> <p>auch bei <code>body</code> Objekt</p>
<code>onrepeat</code>	<p>erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes</p> <p>nicht erzeugt beim Start des aktiven Elementes (siehe <code>onbegin</code>), also des ersten Durchlaufes, der keine Wiederholung ist</p> <p>nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für <code>onrepeat</code> kodiert hat (kein Heraufreichen des Ereignisses <code>onrepeat</code> zum Elternelement)</p> <p><code>.repeatCount</code> bzw. <code>.repeat</code> muss > 1 sein:</p> <p><code>onrepeat</code> wird also <code>.repeatCount - 1</code> mal erzeugt</p> <p>Kind eines Elementes</p> <p>siehe <code>onend</code>, <code>onbegin</code>, <code>.repeatCount</code> und <code>.repeat</code></p>
<code>onreset</code>	<p>erzeugt wenn Element zurückgesetzt wurde per Methode <code>.resetElement()</code>, also die Timeline des Elementes den aktuellen Wert laut <code>.begin</code> wieder erreicht und damit zurückgesetzt wurde</p> <p>also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes</p>
<code>onresume</code>	<p>erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird</p> <p>auch für <code>body</code> Objekt</p>
<code>onreverse</code>	<p>erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also <code>.repeatCount > 0</code> ist)</p> <p><code>.autoReverse</code> muss auf "true" stehen</p>



4.3.2.2.4.3.39.23.9.3.2. *.style.time2.animateColor Behavior-Objekt des Internet Explorer*

Timerobjekt, das zur Farbanimation eines Elementes dient. Die zu animierende Farbart ist im STYLE-Attribut des Elementes **und** im ATTRIBUTENAME-Attribut des Timers zu kodieren. Es handelt sich also um eine Animation mit einer speziellen Style-Eigenschaft, zur Farbe.

Syntax:

XML t:ANIMATECOLOR
Script animateColor

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>

<t:PAR ID="ID_Par"
      BEGIN="0"
      DUR="10"
      FILL="hold"
>
  <t:ANIMATECOLOR ID="ID_Animatecolor1"
    TARGETELEMENT="ID_Div1"
    ATTRIBUTENAME="background-color"
    VALUES="#0000FF;cyan"
    BEGIN="0"
    DUR="5"
    FILL="hold"
  >
  </t:ANIMATECOLOR>
  <t:ANIMATECOLOR ID="ID_Animatecolor2"
    TARGETELEMENT="ID_Div2"
    ATTRIBUTENAME="background-color"
    VALUES="cyan ;#0000FF"
    BEGIN="0"
    DUR="5"
    FILL="hold"
  >
  </t:PAR>
  <DIV ID="ID_Div1"
    CLASS="time_line_Klasse"
    STYLE="position: absolute; left: 68px; width: 279px; top: 260px; height: 217px;
      border: 1px solid black; background-color: green;
    "
  >
    animierter Div
  </DIV>
  <DIV ID="ID_Div2"
    CLASS="time_line_classe"
    STYLE="position: absolute; left: 112px; width: 188px; top: 318px; height: 98px;
      padding-left: 3; background-color: gray;
    "
  >
    animierter Div
  </DIV>
</BODY>
</HTML>
```

Eigenschaften:

.accelerate Beschleunigung des Elementes auf der Timeline
hat keinen Einfluss auf die Dauer der Timeline
auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur
Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten
wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
siehe .decelerate

.accumulate Kumulative Animation eines Elementes auf der Timeline
Element darf nicht aktiv sein, wenn .accumulate definiert werden soll:
Element erst danach starten.
Es kann jede numerische Style-Eigenschaft kumulativ verändert werden und damit
die kumulative Animation des Elementes in der Style-Eigenschaft erzeugen.
Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des
Wertes sein (mathematische Berechnungen erst nach Entfernung der Einheit
per Stringoperationen möglich)
Name der Eigenschaft laut .attributName



Änderung des Wertes pro Durchlauf bzw. kumulativ um den Maximalwert laut `.by`
 Anzahl der Durchläufe bzw. der Wert-Kumulationen laut `.repeatCount`
 Kumulationsschrittweite laut `.calcMode`
 Ereignis des Startes eines Durchlaufes bzw. Kumulation um Maximalwert ist `onrepeat`.
 Wert der Eigenschaft `.repeatCount` muss > 1 sein

Kumulation nicht aktiv:

Nach jedem Durchlauf laut `.repeatCount` erfolgt Rücksetzen der Style-Eigenschaft des Elementes auf den Zustand vor dem Durchlaufbeginn.
 Es wird also die Style-Eigenschaft nicht wertmäßig kumuliert.
 Das Element wird also laut `.repeatCount` mehrmals animiert.
 Pro Durchlauf wird für die Eigenschaft laut `.attributeName` der Wert
 ab Startwert
 in Schrittweite laut `.calcMode`
 um den maximalen Wert laut `.by`
 erhöht.

Kumulation aktiv:

Die Style-Eigenschaft des Elementes wird pro Durchlauf wertmäßig laut
`.repeatCount` um den Wert laut `.by` kumuliert.
 Kumulationsschrittweite laut `.calcMode`
 Das Element wird also genau 1 mal animiert.
 Mit Beginn des Startes des Elementes wird für die Eigenschaft laut `.attributeName`
 der Wert ab Startwert
 in Schrittweite laut `.calcMode`
 um den maximalen Wert aus dem Produkt aus
`.by` mal Anzahl der Wiederholungen
 z.B. laut `.repeatCount`
 erhöht.
 Es wird also über **alle** Wiederholungen der Wert kumuliert.

`.additive`

siehe `.additive .calcMode`
 numerische Werte von Eigenschaften eines Elementes ersetzen oder zu den numerischen Werten der
 gleichnamigen Eigenschaften anderer Elemente addieren
 während der Elemente-Animation auf der Timeline
 Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des Wertes sein
 gültige Eigenschaften sind z.B. `.values` oder `.by`
 sinnvoll für Kombinationen von Elementen (auch bei Wiederholung eines Elementes)
 Element darf nicht aktiv sein, wenn `.additive` definiert werden soll:
 Element erst danach starten.

Achtung: Im Objekt `animatecolor` (t:ANIMATECOLOR) ist `.values` auch als Farbliste definiert !
 siehe `.accumulate .by .attributeName`

`.attributeName`

Bezeichner einer Style-Eigenschaft, also Name eines Attributes, für die Animation anhand
 dieses Attributes auf der Timeline

`.autoReverse`

siehe `.accumulate .by .additive .calcMode`
 automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation
 des Elementes auf der Timeline

`.begin`

Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft `.timeAction`

`.by`

Wert um den die Werterhöhung bei Elemente-Animation(en) auf der Timeline
 per `.additive` oder `.accumulate`
 erfolgen soll

Wert der Eigenschaft:

numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des Wertes sein
 wird für Animation in Einheiten zerlegt laut Schrittweite laut `.calcMode`
 für die Objekte `animate`, `animateMotion` und `animateColor` gilt:
`.by` wird von `.path`, `.to` und `.values` überschrieben

Achtung: Im Objekt `animatecolor` (t:ANIMATECOLOR) ist `.values` auch als Farbliste definiert !
 siehe `.attributeName .accumulate .additive .calcMode`

`.calcMode`

Schrittweite der Werterhöhung bzw. Art der Animation in der Ebene (1D oder 2D)
 bei Elemente-Animation(en) auf der Timeline
 per `.additive` oder `.accumulate`

`.decelerate`

Achtung: Im Objekt `animatecolor` (t:ANIMATECOLOR) ist `.values` auch als Farbliste definiert !
 Verlangsamung des Elementes auf der Timeline
 hat keinen Einfluss auf die Dauer der Timeline
 auch wirksam bei Wiederholungen per Attribute `.repeatCount` oder `.repeatDur`
 Summe der Werte der Attribute `.accelerate` und `.decelerate` darf nicht 1 überschreiten
 wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
 siehe `.accelerate`

`.dur`

Dauer Objektaktivitäten laut Eigenschaft `.timeAction`

`.end`

alternativ: Eigenschaft `.end`

Objektaktivitäten laut Eigenschaft `.timeAction` beenden
 ab IE 6.x

`.fill`

alternativ: Eigenschaft `.dur`

Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber
 bevor die Timeline des Elternelementes endet
 ist Ersatz für die Eigenschaft `endHold`, die deprecated ist und nicht mehr verwendet
 werden darf !!



<code>.from</code>	Startwert zur Werterhöhung bei Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> für die Objekte <code>animate</code> , <code>animateMotion</code> und <code>animateColor</code> gilt: <code>.from</code> wird von <code>.path</code> und <code>.values</code> überschrieben Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert !
<code>.hasMedia</code>	Objekt ist HTML-Media-Objekt
<code>.keySplines</code>	Wert eines Pixel-Intervalles zur Werterhöhung bei 2D-Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> benötigt <code>.calcMode</code> auf "spline" <code>.keyTimes</code> <code>.values</code> oder <code>.path</code> Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert !
<code>.keyTimes</code>	Zeitwert eines Pixel-Intervalles zur Werterhöhung bei 2D-Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> benötigt <code>.calcMode</code> auf "spline" <code>.keySplines</code> <code>.values</code> oder <code>.path</code> Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert !
<code>.repeatCount</code>	aktuelle Nummer der Wiederholung bei Loop
<code>.repeatDur</code>	Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline verlangt kodierte Eigenschaft <code>.dur</code> oder <code>.repeat</code> darf nicht kodiert werden mit der Eigenschaft <code>.repeatCount</code>
<code>.restart</code>	generelle Restartmöglichkeit eines Elementes auf der Timeline ersetzt die Eigenschaft <code>eventRestart</code> , da diese deprecated ist und nicht mehr verwendet werden darf !!
<code>.speed</code>	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
<code>.systemBitrate</code>	wird hier nicht erklärt
<code>.systemCaptions</code>	wird hier nicht erklärt
<code>.systemLanguage</code>	Sprache festlegen für das Objekt
<code>.systemOverdubOrSubtitle</code>	wird hier nicht erklärt
<code>.targetElement</code>	ID des zu animierenden Elementes auf der Timeline muss für Kindelement immer kodiert werden, wenn nicht das Elternelement animiert werden soll muss nicht kodiert werden, wenn kein Elternelement vorliegt Achtung: Objekt <code>body</code> ist das oberste Elternelement im Dokument und somit haben Elemente innerhalb BODY immer Eltern Empfehlung: immer kodieren ist der Bezug des Time-Container auf das zu animierende Element
<code>.timeContainer</code>	Typ der Timeline des Objektes
<code>.timeParent</code>	Zeiger auf das Eltern-Timeline
<code>.to</code>	Endwert zur Werterhöhung bei Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> für die Objekte <code>animate</code> , <code>animateMotion</code> und <code>animateColor</code> gilt: <code>.to</code> wird von <code>.path</code> und <code>.values</code> überschrieben <code>.to</code> überschreibt <code>.by</code> Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert !
<code>.values</code>	2D-Animation bei der Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> anhand absoluter Koordinaten im Grafiksystem siehe <code>.path</code> für Vektorgraphik-Animation anhand realtiver und absoluter Koordinaten Werteliste korrespondiert zu der Werteliste der Eigenschaft <code>.keyTimes</code> benötigt <code>.calcMode</code> auf "spline" <code>.keySplines</code> und <code>.keyTimes</code> für die Objekte <code>animate</code> , <code>animateMotion</code> und <code>animateColor</code> gilt: <code>.values</code> wird von <code>.path</code> überschrieben <code>.values</code> überschreibt <code>.by</code> <code>.from</code> <code>.to</code> Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert !
<code>.values</code>	Farbwerte für die Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> nur für Objekt <code>animatecolor</code> (t:ANIMATECOLOR) Achtung: Nur im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert ! Es existieren als 2 kodierbare Varianten von <code>.values</code> .

Methoden:

<code>.activeTimeToParentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.activeTimeToSegmentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.beginElement()</code>	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes.. per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.beginElementAt()</code>	Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort,



	also ohne Wartezeit
	wenn zusätzlich ein weiterer Start (oder mehrere Starts)
	mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert
	mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt
	also Neustart nach einem erfolgten Start
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.documentTimeToParentTime()	Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
.endElement()	aktives Element auf der Timeline stoppen
	identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde
	Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.endElementAt()	aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen
	sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes
	wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.parentTimeToActiveTime()	Wert der Timeline der Eltern zum korrespondierenden Wert der aktives Timeline des Elementes konvertieren
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.parentTimeToDocumentTime()	Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
.pauseElement()	aktives Element auf Timeline pausieren lassen
	ersetzt Methode pause(), die deprecated ist und nicht mehr verwendet werden darf !
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.resetElement()	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
.resumeElement()	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben
	falls Element nicht pausiert, passiert nichts
	ersetzt die Methode resume(), da sie deprecated ist und nicht mehr verwendet werden darf
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.seekActiveTime()	aktives Element animieren ab einem Zeitpunkt auf der Timeline
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	alle Media-Typen für Element zulässig
	siehe Eigenschaft .canSeek
.seekSegmentTime()	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline
	ohne Wiederholung der Animation
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.seekTo()	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline
	einschliesslich möglicher Wiederholungen der Animation
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.seekToFrame()	Frame eines aktiven Elementes auf der Timeline anwählen
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.segmentTimeToActiveTime()	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
.segmentTimeToSimpleTime()	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
.simpleTimeToSegmentTime()	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren

Events:

onbegin	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird
	nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Element
	erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Eltern-Element
	siehe onend und onrepeat
onend	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive



	aller Wiederholungen laut <code>.repeatCount</code> bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit des Kindelement ebenfalls endet nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code> siehe <code>onbegin</code> und <code>onrepeat</code>
onpause	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt
onrepeat	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe <code>onbegin</code>), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für <code>onrepeat</code> kodiert hat (kein Heraufreichen des Ereignisses <code>onrepeat</code> zum Elternelement) <code>.repeatCount</code> bzw. <code>.repeat</code> muss > 1 sein: <code>onrepeat</code> wird also <code>.repeatCount - 1</code> mal erzeugt Kind eines Elementes
onreset	siehe <code>onend</code> , <code>onbegin</code> , <code>.repeatCount</code> und <code>.repeat</code> erzeugt wenn Element zurückgesetzt wurde per Methode <code>.resetElement()</code> , also die Timeline des Elementes den aktuellen Wert laut <code>.begin</code> wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanzierung des Elementes
onresume	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
onreverse	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also <code>.repeatCount > 0</code> ist) <code>.autoReverse</code> muss auf "true" stehen

4.3.2.2.4.3.39.23.9.3.3. *.style.time2.animateMotion Behavior-Objekt des Internet Explorer*

Timerobjekt, das zur 2D-Animation (in der Ebene) des Elementes dient.

Es können nur Style-Eigenschaften des Elementes dazu verwendet werden, die die räumliche Lage des Elementes angeben wie z.B. `top` oder `left`.

Zugleich ist in STYLE-Attribut des Elementes zu kodieren, ob die Lage des Elementes relativ oder absolut ist:

Bsp. `STYLE="position:absolute; ..."`

Die Kodierung der neuen Positionen muss im Timer mit mindestens einem der folgenden Attributen erfolgen:

`.begin`
`.by`
`.from`
`.to`

Wert eines jeden dieser Attribute:

String als Liste aus kommagetrennten Werten

Listenelement: Wert, der zum Typ zur Style-Eigenschaft passen muss

Folge der Listenelemente (Wertfolge):

entspricht der Folge der Style-Eigenschaften im Style-Attribut

Bsp.:

`STYLE="position:absolute; top:100px; left:50px; width:100px; height:50px;"`

`TO="200,300"`

also `top` auf 200
`left` auf 300

nicht animiert werden `width` und `height`

Analog gilt das bei Verwendung von `.keySplines`, `.keyTimes` und `.values`

Syntax:

XML `t:ANIMATEMOTION`
 Script `animateMotion`

Beispiel 1:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT LANGUAGE="JScript">
function HochZaehlen()
{
```



```

        ID_Div1.innerText = "Zähler: "
                          + (ID_Animation.currTimeState.repeatCount + 1);
    }
</SCRIPT>
</HEAD>
<BODY>
    <DIV ID="ID_Div1"></DIV>
    <DIV ID="ID_Div2"
        CLASS="time_line_klasse"
        STYLE="position:absolute; left:50px; top:250px; width:100px; height:100px;"
    >
    </DIV>
    <t:ANIMATEMOTION
        ID="ID_Animation"
        TARGETELEMENT="ID_Div2"
        TO="150,0"
        DUR="1"
        AUTOREVERSE="true"
        REPEATCOUNT="5"
        onrepeat="HochZachlen()"
    >
    </t:ANIMATEMOTION>
</BODY>
</HTML>

```

Beispiel 2:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
    .div_Klasse{position:absolute; top:195px; height:100px; width:150px; border:solid black 2px;}
</STYLE>
</HEAD>
<BODY>
    <SPAN ID="ID_Span1"
        CLASS="time_line_klasse"
        DUR=".1"
        REPEATCOUNT="indefinite"
        FILL="hold"
        onrepeat="innerText=( ID_Animatemotion.currTimeState.simpleTime);"
    >
        0
    </SPAN>
    <BR>
    <SPAN ID=" ID_Span2"
        CLASS="time_line_klasse"
        DUR=".1"
        REPEATCOUNT="indefinite"
        FILL="hold"
        onrepeat="innerText=( ID_Animatemotion.currTimeState.activeTime);"
    >
        0
    </SPAN>
    <DIV ID="ID_Div"
        CLASSE="div_klasse"
    >
        animierter Div
    </DIV>
    <t:ANIMATEMOTION ID="ID_Animatemotion"
        BEGIN="1; ID_Button.click+1"
        DUR="6s"
        AUTOREVERSE="true"
        CALCMODE="spline"
        KEYSPLINES="0 1 .5 1;.5 1 0 1"
        KEYTIMES="0;.5;1"
        VALUES="25,0;250,50;500,0"
        TARGETELEMENT="ID_Div"
        FILL="hold"
    >
    </t:ANIMATEMOTION>
    <BUTTON ID="ID_Button">Restart</BUTTON>
</BODY>
</HTML>

```



Beispiel 3:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<DIV ID="ID_Div1"
CLASS="time_line_klasse"
STYLE="position:absolute;top:50px;left:50px;height:100px;width:150px;
border:solid black 1px;
"
>
animierter Div mit Vektorgraphik
</DIV>
<t:ANIMATEMOTION ID="ID_Animatemotion1"
TARGETELEMENT="ID_Div1"
BEGIN="ID_Button.click"
PATH="M 0 0 L 100 0 v 100 h -100 V 0"
DUR="3"
>
</t:ANIMATEMOTION>
<DIV ID="ID_Div2"
CLASS="time_line_klasse"
STYLE="position:absolute;top:50px;left:250px;height:100px;width:150px;
border:solid black 1px;
"
>
animierter Div mit Vektorgraphik
</DIV>
<t:ANIMATEMOTION ID="ID_Animatemotion2"
TARGETELEMENT="ID_Div2"
BEGIN="ID_Button.click"
PATH="m 0 0 L 100 0 100 100 0 100 z"
DUR="3"
>
</t:ANIMATEMOTION>
<BUTTON ID="ID_Button">Starten/Restarten</BUTTON>
</BODY>
</HTML>

```

Beispiel 4:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
window.onload = Rekursion; // ohne () kodieren !!

function Rekursion()
{window.setInterval(Anzeige, 100); }

function Anzeige()
{
ID_Span1.innerHTML = "&nbsp;activeTimeToSegmentTime:&nbsp;";
+ (ID_Animate.activeTimeToSegmentTime(
ID_Div.currTimeState.activeTime
));

ID_Span2.innerHTML = "&nbsp;segmentTime:&nbsp;";
+ (ID_Animate.currTimeState.segmentTime);
}
</SCRIPT>
</HEAD>
<BODY>
<SPAN ID="ID_Span0"
CLASS="time_line_klasse"
DUR="1"

```



```

        REPEATCOUNT="indefinite"
        onrepeat="innerText=parseInt(document.body.currTimeState.activeTime);"
    >
        0
    </SPAN>
    <DIV ID="ID_Div"
        CLASS="time_line_klasse"
        STYLE="position:absolute; left:50px; top:250px; width:100px; height:100px;"
    >
    </DIV>
    <t:ANIMATEMOTION ID="ID_Animatemotion"
        TARGETELEMENT="ID_Div"
        TO="250,0"
        DUR="3"
        AUTOREVERSE="true"
    >
    </t:ANIMATEMOTION>
    <SPAN ID="ID_Span1">
        activeTimeToSegmentTime:
    </SPAN>
    <BR>
    <SPAN ID="ID_Span2">
        segmentTime:
    </SPAN>
</BODY>
</HTML>

```

Eigenschaften:

.accelerate
Beschleunigung des Elementes auf der Timeline
hat keinen Einfluss auf die Dauer der Timeline
auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur
Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten
wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
siehe .decelerate

.accumulate
Kumulative Animation eines Elementes auf der Timeline
Element darf nicht aktiv sein, wenn .accumulate definiert werden soll:
Element erst danach starten.
Es kann jede numerische Style-Eigenschaft kumulativ verändert werden und damit
die kumulative Animation des Elementes in der Style-Eigenschaft erzeugen.
Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des
Wertes sein (mathematische Berechnungen erst nach Entfernung der Einheit
per Stringoperationen möglich)
Name der Eigenschaft laut .attributName
Änderung des Wertes pro Durchlauf bzw. kumulativ um den Maximalwert laut .by
Anzahl der Durchläufe bzw. der Wert-Kumulationen laut .repeatCount
Kumulationsschrittweite laut .calcMode
Ereignis des Startes eines Durchlaufes bzw. Kumulation um Maximalwert ist onrepeat.
Wert der Eigenschaft .repeatCount muss > 1 sein
Kumulation nicht aktiv:
Nach jedem Durchlauf laut .repeatCount erfolgt Rücksetzen der Style-Eigenschaft des
Elementes auf den Zustand vor dem Durchlaufbeginn.
Es wird also die Style-Eigenschaft nicht wertmäßig kumuliert.
Das Element wird also laut .repeatCount mehrmals animiert.
Pro Durchlauf wird für die Eigenschaft laut .attributName der Wert
ab Startwert
in Schrittweite laut .calcMode
um den maximalen Wert laut .by
erhöht.
Kumulation aktiv:
Die Style-Eigenschaft des Elementes wird pro Durchlauf wertmäßig laut
.repeatCount um den Wert laut .by kumuliert.
Kumulationsschrittweite laut .calcMode
Das Element wird also genau 1 mal animiert.
Mit Beginn des Startes des Elementes wird für die Eigenschaft laut .attributName
der Wert ab Startwert
in Schrittweite laut .calcMode
um den maximalen Wert aus dem Produkt aus
.by mal Anzahl der Wiederholungen
z.B. laut .repeatCount
erhöht.
Es wird also über **alle** Wiederholungen der Wert kumuliert.
siehe .additive .calcMode

.additive
numerische Werte von Eigenschaften eines Elementes ersetzen oder zu den numerischen Werten der
gleichnamigen Eigenschaften anderer Elemente addieren
während der Elemente-Animation auf der Timeline



	<p>Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des Wertes sein gültige Eigenschaften sind z.B. <code>.values</code> oder <code>.by</code> sinnvoll für Kombinationen von Elementen (auch bei Wiederholung eines Elementes) Element darf nicht aktiv sein, wenn <code>.additive</code> definiert werden soll: Element erst danach starten.</p> <p>Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert ! siehe <code>.accumulate</code> <code>.by</code> <code>.attributeName</code></p>
<code>.autoReverse</code>	automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
<code>.begin</code>	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft <code>.timeAction</code>
<code>.by</code>	Wert um den die Werterhöhung bei Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> erfolgen soll
	<p>Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des Wertes sein wird für Animation in Einheiten zerlegt laut Schrittweite laut <code>.calcMode</code> für die Objekte <code>animate</code>, <code>animateMotion</code> und <code>animateColor</code> gilt: <code>.by</code> wird von <code>.path</code>, <code>.to</code> und <code>.values</code> überschrieben</p> <p>Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert ! siehe <code>.attributeName</code> <code>.accumulate</code> <code>.additive</code> <code>.calcMode</code></p>
<code>.calcMode</code>	Schrittweite der Werterhöhung bzw. Art der Animation in der Ebene (1D oder 2D) bei Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code>
<code>.decelerate</code>	<p>Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert ! Verlangsamung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute <code>.repeatCount</code> oder <code>.repeatDur</code> Summe der Werte der Attribute <code>.accelerate</code> und <code>.decelerate</code> darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet</p> <p>siehe <code>.accelerate</code></p>
<code>.dur</code>	Dauer Objektaktivitäten laut Eigenschaft <code>.timeAction</code> alternativ: Eigenschaft <code>.end</code>
<code>.end</code>	Objektaktivitäten laut Eigenschaft <code>.timeAction</code> beenden ab IE 6.x alternativ: Eigenschaft <code>.dur</code>
<code>.fill</code>	Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber bevor die Timeline des Elternelementes endet ist Ersatz für die Eigenschaft <code>endHold</code> , die deprecated ist und nicht mehr verwendet werden darf !!
<code>.from</code>	Startwert zur Werterhöhung bei Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> für die Objekte <code>animate</code> , <code>animateMotion</code> und <code>animateColor</code> gilt: <code>.from</code> wird von <code>.path</code> und <code>.values</code> überschrieben
<code>.hasMedia</code>	Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert !
<code>.keySplines</code>	Objekt ist HTML-Media-Objekt
Timeline	Wert eines Pixel-Intervall zur Werterhöhung bei 2D-Elemente-Animation(en) auf der per <code>.additive</code> oder <code>.accumulate</code> benötigt <code>.calcMode</code> auf "spline" <code>.keyTimes</code> <code>.values</code> oder <code>.path</code>
<code>.keyTimes</code>	<p>Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert ! Zeitwert eines Pixel-Intervall zur Werterhöhung bei 2D-Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> benötigt <code>.calcMode</code> auf "spline" <code>.keySplines</code> <code>.values</code> oder <code>.path</code></p>
<code>.origin</code>	<p>Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert ! Bezugspunkt der Animation eines Elementes auf der Timeline nur für Objekt <code>animatemotion</code> (t:ANIMATEMOTION) und <code>.additive</code> muss "replace" sein</p>
<code>.path</code>	<p>Liste von Werten für 2D-Vektorgraphik-Animation eines Elementes auf der Timeline nur für Objekt <code>animatemotion</code> (t:ANIMATEMOTION) siehe <code>.values</code> für 2D-Animation anhand absoluter Koordinaten im Grafiksystem wenn <code>.calcMode</code> auf "paced" so werden Move To-Kommandos ignoriert für die Objekte <code>animate</code>, <code>animateMotion</code> und <code>animateColor</code> gilt: <code>.path</code> wird von keiner Eigenschaft überschrieben <code>.path</code> überschreibt <code>.by</code> <code>.from</code> <code>.to</code> <code>.values</code></p>
<code>.repeatCount</code>	Achtung: Im Objekt <code>animatecolor</code> (t:ANIMATECOLOR) ist <code>.values</code> auch als Farbliste definiert !
<code>.repeatDur</code>	aktuelle Nummer der Wiederholung bei Loop Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline verlangt kodierte Eigenschaft <code>.dur</code> oder <code>.repeat</code> darf nicht kodiert werden mit der Eigenschaft <code>.repeatCount</code>
<code>.restart</code>	generelle Restartmöglichkeit eines Elementes auf der Timeline



	ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!
.speed	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.targetElement	ID des zu animierenden Elementes auf der Timeline muss für Kindelement immer kodiert werden, wenn nicht das Elternelement animiert werden soll muss nicht kodiert werden, wenn kein Elternelement vorliegt Achtung: Objekt body ist das oberste Elternelement im Dokument und somit haben Elemente innerhalb BODY immer Eltern Empfehlung: immer kodieren ist der Bezug des Time-Container auf das zu animierende Element
.timeContainer	Typ der Timeline des Objektes
.timeParent	Zeiger auf das Eltern-Timeline
.to	Endwert zur Werterhöhung bei Elemente-Animation(en) auf der Timeline per .additive oder .accumulate für die Objekte animate, animateMotion und animateColor gilt: .to wird von .path und .values überschrieben .to überschreibt .by
.values	Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert ! 2D-Animation bei der Elemente-Animation(en) auf der Timeline per .additive oder .accumulate anhand absoluter Koordinaten im Grafiksystem siehe .path für Vektorgraphik-Animation anhand realtiver und absoluter Koordinaten Werteliste korrespondiert zu der Werteliste der Eigenschaft .keyTimes benötigt .calcMode auf "spline" .keySplines und .keyTimes für die Objekte animate, animateMotion und animateColor gilt: .values wird von .path überschrieben .values überschreibt .by .from .to Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert !
Methoden:	
.activeTimeToParentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.activeTimeToSegmentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.beginElement()	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.beginElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start per Eigenschaft .isActive das Element auf Aktivsein prüfen
.documentTimeToParentTime()	Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
.endElement()	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.endElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne per Eigenschaft .isActive das Element auf Aktivsein prüfen
.parentTimeToActiveTime()	Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.parentTimeToDocumentTime()	Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
.pauseElement()	aktives Element auf Timeline pausieren lassen ersetzt Methode pause(), die deprecated ist und nicht mehr verwendet werden darf ! per Eigenschaft .isActive das Element auf Aktivsein prüfen



<code>.resetElement()</code>	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
<code>.resumeElement()</code>	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben falls Element nicht pausiert, passiert nichts ersetzt die Methode <code>resume()</code> , da sie deprecated ist und nicht mehr verwendet werden darf per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.seekActiveTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Timeline wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen alle Media-Typen für Element zulässig siehe Eigenschaft <code>.canSeek</code>
<code>.seekSegmentTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline ohne Wiederholung der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.seekTo()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.seekToFrame()</code>	Frame eines aktiven Elementes auf der Timeline anwählen wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.segmentTimeToActiveTime()</code>	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
<code>.segmentTimeToSimpleTime()</code>	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
<code>.simpleTimeToSegmentTime()</code>	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren

Events:

<code>onbegin</code>	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Element erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Eltern-Element siehe <code>onend</code> und <code>onrepeat</code>
<code>onend</code>	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut <code>.repeatCount</code> bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit das Kindelement ebenfalls endet nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code> siehe <code>onbegin</code> und <code>onrepeat</code>
<code>onpause</code>	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt
<code>onrepeat</code>	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe <code>onbegin</code>), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für <code>onrepeat</code> kodiert hat (kein Heraufreichen des Ereignisses <code>onrepeat</code> zum Elternelement) <code>.repeatCount</code> bzw. <code>.repeat</code> muss > 1 sein: onrepeat wird also <code>.repeatCount - 1</code> mal erzeugt Kind eines Elementes siehe <code>onend</code> , <code>onbegin</code> , <code>.repeatCount</code> und <code>.repeat</code>
<code>onreset</code>	erzeugt wenn Element zurückgesetzt wurde per Methode <code>.resetElement()</code> , also die Timeline des Elementes den aktuellen Wert laut <code>.begin</code> wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
<code>onresume</code>	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
<code>onreverse</code>	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird



(auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist)
 .autoReverse muss auf "true" stehen

4.3.2.2.4.3.39.23.9.3.4. *.style.time2.animation Behavior-Objekt des Internet Explorer*

Timerobjekt, das zur Animation eines beliebigen Elementes dient

Media-Element erwartet Datenstrom aus Daten, der meist in einer externen Datei abgelegt ist z.B. Gif-Datei oder Video oder MP3

Spezielle Timerobjekte existieren zusätzlich

z.B. für Audio t:AUDIO
 Bild t:IMG,
 Media t:MEDIA
 Video t:VIDEO
 MP3 playItem Objekt in Verbindung mit z.B. t:MEDIA

die direkt auf das Element zugeschnitten sind und mehr Funktionen unterstützen (siehe Objektbeschreibung)

Syntax:

XML t:ANIMATION Element
 Script animation

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:IMG ID="ID_Img"
SRC="test.gif"
STYLE="position:relative;top:25px;left:50px;height:100px;width:100px;"
onmediaerror="alert(' Datei test.gif nicht ladbar !') "
onrepeat="alert(' Aktuelle Wiederholung: ' + ID_Animation.currTimeState.repeatCount);"
>
</t:IMG>
<t:ANIMATION ID="ID_Animation"
TARGETELEMENT="ID_Img"
TO="0,400"
DUR="2"
BEGIN="0"
ACCELERATE="1"
AUTOREVERSE="true"
REPEATCOUNT="5"
>
</t:ANIMATION>
</BODY>
</HTML>
```

Eigenschaften:

.abstract	Beschreibung einer Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei wird der Text unter ABSTRACT vom aktuellen Eintrag geliefert und nicht der Datei selbst
.accelerate	Beschleunigung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .decelerate
.author	Name des Autor der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von AUTHOR des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.autoReverse	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.begin BOUNDARY	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.bufferingProgress	aktueller prozentualer Status des Pufferens eines Datenflusses (Data Stream) einer Media-Datei auf der Timeline Prozentwert des bisher erfolgten Pufferens nur für Media-Datei mit Datenfluss
.canPause	generelle Pausierungsmöglichkeit einer Media-Datei auf der Timeline
.canSeek	generelle Möglichkeit der Auswahl eines Zeitpunktes einer Media-Datei auf der Timeline Auswahl per Seek-Methoden .seekActiveTime()



	<code>.seekSegmentTime()</code>
	<code>.seekTo()</code>
	<code>.seekToFrame()</code>
	nicht jeder Media-Typ unterstützt Seek
<code>.clipBegin</code>	Startzeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline
	Eigenschaft <code>.canSeek</code> muss true liefern
	nicht jeder Media-Typ unterstützt Clipping
<code>.clipEnd</code>	Endezeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline
	Eigenschaft <code>.canSeek</code> muss true liefern
	nicht jeder Media-Typ unterstützt Clipping
<code>.copyright</code>	Copyright der Media-Datei auf der Timeline
	bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von COPYRIGHT des aktiven Eintrages geliefert und nicht den der Datei.
	Track entspricht <code>playItem</code> Objekt
	Trackliste liegt in der Behavior-Collection <code>.style.time2.playList</code> als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
<code>.currentFrame</code>	Nummer des aktuellen Frame einer Media-Datei auf der Timeline
	nicht alle Media-Typen unterstützen Frames
<code>.decelerate</code>	Verlangsamung des Elementes auf der Timeline
	hat keinen Einfluss auf die Dauer der Timeline
	auch wirksam bei Wiederholungen per Attribut <code>.repeatCount</code> oder <code>.repeatDur</code>
	Summe der Werte der Attribute <code>.accelerate</code> und <code>.decelerate</code> darf nicht 1 überschreiten
	wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
	siehe <code>.accelerate</code>
<code>.downloadCurrent</code>	Anzahl der bisher downgeloadeten Bytes beim Laden einer Media-Datei auf der Timeline
	(Laden des Daten-Stream in Form der Media-Datei)
	nur für Media-Datei mit Datenfluss
<code>.downloadTotal</code>	Anzahl der insgesamt downgeloadeten Bytes nach dem kompletten Laden einer Media-Datei auf der Timeline
	(Laden des Daten-Stream in Form der Media-Datei)
<code>.dur</code>	Dauer Objektaktivitäten laut Eigenschaft <code>.timeAction</code>
	alternativ: Eigenschaft <code>.end</code>
<code>.end</code>	Objektaktivitäten laut Eigenschaft <code>.timeAction</code> beenden
	ab IE 6.x
	alternativ: Eigenschaft <code>.dur</code>
<code>.fill</code>	Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber
	bevor die Timeline des Elternelementes endet
	ist Ersatz für die Eigenschaft <code>endHold</code> , die deprecated ist und nicht mehr verwendet werden darf !!
<code>.hasAudio</code>	Media-Datei mit Audioinhalt auf der Timeline
	nicht alle Media-Typen können Audio beinhalten
	Stummschaltung oder Lautstärkeeinstellungen sind dabei egal
<code>.hasDownloadProgres</code>	Start des Donloades einer Media-Datei auf der Timeline
<code>.hasMedia</code>	Objekt ist HTML-Media-Objekt
<code>.hasVisual</code>	Media-Datei mit visuellem Inhalt auf der Timeline
	visuelle Daten werden auf dem Bildschirm sichtbar
	nicht alle Media-Typen können sichtbare Daten beinhalten
IMMEDIATEEND	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
<code>.isStreamed</code>	Media-Datei mit Datenfluss (Data Stream) auf der Timeline
	nicht alle Media-Typen unterstützen Datenstrom
<code>.latestMediaTime</code>	Wartezeit für Start der Wiedergabe eines Media-Elementes auf der Timeline
	nicht für Media-Datei mit Datenfluss
LONGTRANSITION	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
<code>.mediaDur</code>	Länge der Media-Datei in Sekunden (Länge bei der Wiedergabe) auf der Timeline
	Ereignis <code>onmediacomplete</code> muss ausgelöst also die Media-Datei komplett geladen worden sein
<code>.mediaHeight</code>	aktuelle Höhe des Media-Elementes in Pixels auf der Timeline
	Standardgemäß wird die Elementhöhe aus der Höhenangabe in der Media-Datei verwendet
	(ansonsten STYLE-Angabe zu <code>.style.height</code> im Element kodieren)
<code>.mediaWidth</code>	aktuelle Breite des Media-Elementes in Pixels auf der Timeline
	Standardgemäß wird die Elementhöhe aus der Breitenangabe in der Media-Datei verwendet
	(ansonsten STYLE-Angabe zu <code>.style.width</code> im Element kodieren)
<code>.mimeType</code>	MIME-Typ (Multipurpose Internet Mail Extension-Typ) des Media-Elementes auf der Timeline
MODULATE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
MOTIFNAME	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
<code>.mute</code>	Audio aktiv oder aus (stumm) auf der Timeline
	wenn Eigenschaft für <code>body</code> Objekt kodiert, so alle Audio-Elemente im Body davon betroffen
	beeinflusst nicht die Systemlautstärke-Regelung von Windows oder anderen Soundmixern
	siehe Objekt <code>bgsound</code>
<code>.player</code>	Wiedergabeplayer einem Media-Element zuordnen für Wiedergabe auf der Timeline
	Player ist eine Softwarekomponente, die im System und im Browser installiert sein muss
	ab IE 6.x werden keine Plugins mehr unterstützt sondern nur noch ActiveX-Controls
	z.B. ist {6BF52A52-394A-11d3-B153-00C04F79FAA6} das ActiveX-Control für den



	Windows Media Player 7.1
.playerObject	Zeiger auf den Wiedergabeplayer, der das Media-Element auf der Timeline wiedergibt besonders zu verwenden, wenn Player über ActiveX-Control im Browser implementiert ist Zeiger ermöglicht, die Methoden und Eigenschaften des Players anzusprechen siehe Playerbeschreibung des Herstellers vom Player (im Falle von Microsoft sind die Methoden und Eigenschaften im jeweiligen SDK der Player-Version zu finden)
.rating	Playerart laut .player
.repeatCount	Rating der Media-Datei auf der Timeline
.repeatDur	aktuelle Nummer der Wiederholung bei Loop
	Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline
	verlangt kodierte Eigenschaft .dur oder .repeat
	darf nicht kodiert werden mit der Eigenschaft .repeatCount
.restart	generelle Restartmöglichkeit eines Elementes auf der Timeline
	ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!
SEGMENTTYPE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.speed	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
.src	Url einer Media-Datei auf der Timeline
.syncBehavior	Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes sinnvoll vorallem bei Time-Container (Gruppen von Objekten)
	Es muss die Eigenschaft .syncmaster kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der Beschreibung zum jeweiligen Objekt zu entnehmen.
	siehe .syncTolerance und .syncmaster
.syncMaster	Synchronisierung der Animation des im Container liegenden Elementes auf Timeline (Container ist Master in der Synchronisierung), wobei ein Master nur genau ein zu synchronisierendes Element haben darf (Eineindeutigkeit).
	nur sinnvoll bei Zwangssynchronisierung (siehe .syncBehavior)
	ersetzt die Eigenschaft clockSource, da diese deprecated ist und nicht mehr verwendet werden darf !
	siehe .syncTolerance und .syncBehavior
.syncTolerance	zeitliche Toleranz in Sekunden für Zwangs-Synchronisation von Elementen auf der Timeline also wenn Eigenschaft .syncBehavior auf "locked" gesetzt ist
	sinnvoll vorallem bei Time-Container (Gruppen von Objekten)
	siehe .syncBehavior und .syncMaster
.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.timeAction	Aktion des Objektes in der Timeline
	Aktion nur ausführbar wenn Objekt aktiv ist UND Timeline aktiv ist
.timeContainer	Typ der Timeline des Objektes
.timeParent	Zeiger auf das Eltern-Timeline
.title	Titel der Media-Datei auf der Timeline
	bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von TITLE des aktiven Eintrages geliefert und nicht den der Datei.
	Track entspricht playItem Objekt
	Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
TRANSITIONTYPE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.type	MIME-Typ eines Media-Elementes auf der Timeline
	Media-Datei zum Element laut Eigenschaft .src
.updateMode	Art des Updates der Eigenschaften eines Elementes nach dem Start des Elementes auf der Timeline
	folgende Eigenschaften können geupdatet werden:
	.autoReverse
	.begin
	.dur
	.end
	.fill
	.repeatCount
	.repeatDur
	.speed
.volume	Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes für alle Audio-Elemente des body Objektes auch möglich
	Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus !
	siehe Objekt bgsound

Methoden:

.activeTimeToParentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.activeTimeToSegmentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.beginElement()	Element auf der Timeline starten, also aktivieren



	identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.beginElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start
.documentTimeToParentTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
.endElement()	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.endElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne
.parentTimeToActiveTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren
.parentTimeToDocumentTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
.pauseElement()	aktives Element auf Timeline pausieren lassen ersetzt Methode pause(), die deprecated ist und nicht mehr verwendet werden darf !
.resetElement()	per Eigenschaft .isActive das Element auf Aktivsein prüfen alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
.resumeElement()	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben falls Element nicht pausiert, passiert nichts ersetzt die Methode resume(), da sie deprecated ist und nicht mehr verwendet werden darf
.seekActiveTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element animieren ab einem Zeitpunkt auf der Timeline wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen
.seekSegmentTime()	alle Media-Typen für Element zulässig siehe Eigenschaft .canSeek aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline ohne Wiederholung der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen
.seekTo()	Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft .canSeek aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen
.seekToFrame()	nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft .canSeek Frame eines aktiven Elementes auf der Timeline anwählen wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen
.segmentTimeToActiveTime()	nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft .canSeek Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
.segmentTimeToSimpleTime()	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
.simpleTimeToSegmentTime()	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren

Events:

onbegin erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline



	aktiv wird
	nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Element
	erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Eltern-Element
	siehe <code>onend</code> und <code>onrepeat</code>
<code>onend</code>	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut <code>.repeatCount</code> bzw. wenn das aktive Element gestoppt wird
	erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit des Kindelement ebenfalls endet
	nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht
	siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code>
	siehe <code>onbegin</code> und <code>onrepeat</code>
<code>onmediacomplete</code>	erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde für Animation per Timeline
	Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc.
	sinnvoll für Start des Elementes auf der Timeline
	siehe <code>onmediaerror</code>
<code>onmediaerror</code>	erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält
	für Animation per Timeline
	Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc.
	sinnvoll für Start des Elementes auf der Timeline
	ersetzt das Ereignis <code>onmedialoadfailed</code> , das deprecated ist und nicht mehr verwendet werden darf !
	siehe <code>onmediacomplete</code>
<code>onoutofsync</code>	erzeugt wenn das Element seine Timeline verliert, also nicht mehr auf seiner Timeline animiert wird (nicht synchron zur Timeline des Elementes)
	sinnvoll nur bei Zwangssynchronisierung per <code>.syncBehavior</code> auf "locked": Eventhandler sollte neu synchronisieren z.B. Reset des Elementes durch <code>.resetElement()</code>
	siehe <code>onsyncstored</code>
<code>onpause</code>	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren
	auch bei body Objekt
<code>onrepeat</code>	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes
	nicht erzeugt beim Start des aktiven Elementes (siehe <code>onbegin</code>), also des ersten Durchlaufes, der keine Wiederholung ist
	nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für <code>onrepeat</code> kodiert hat (kein Heraufreichen des Ereignisses <code>onrepeat</code> zum Elternelement)
	<code>.repeatCount</code> bzw. <code>.repeat</code> muss > 1 sein: onrepeat wird also <code>.repeatCount - 1</code> mal erzeugt
	Kind eines Elementes
	siehe <code>onend</code> , <code>onbegin</code> , <code>.repeatCount</code> und <code>.repeat</code>
<code>onreset</code>	erzeugt wenn Element zurückgesetzt wurde per Methode <code>.resetElement()</code> , also die Timeline des Elementes den aktuellen Wert laut <code>.begin</code> wieder erreicht und damit zurückgesetzt wurde
	also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
<code>onresume</code>	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird
	auch für body Objekt
<code>onreverse</code>	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also <code>.repeatCount > 0</code> ist)
	<code>.autoReverse</code> muss auf "true" stehen
<code>onseek</code>	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde:
	<code>.seekActiveTime()</code>
	<code>.seekSegmentTime()</code>
	<code>.seekTo()</code>
	<code>.seekToFrame()</code>
<code>onsyncstored</code>	erzeugt wenn Element gerade mit seiner Timeline synchronisiert wird nach vorausgegangenem Abbruch der Synchronisation
	siehe <code>onoutofsync</code>
	sinnvoll nur bei Zwangssynchronisierung per <code>.syncBehavior</code> auf "locked"

4.3.2.2.4.3.39.23.9.3.5. *.style.time2.audio Behavior-Objekt des Internet Explorer*

Timerobjekt zur Wiedergabe eines Audio-Elementes im HTML-Dokument

Audio-Element muss nicht mit diesen Objekt wiedergegeben werden, kann aber mit Playlist: siehe Objekt `playItem` und Behavior-Collection `.style.time2.playlist`

Syntax:

XML t:AUDIO
Script audio



Beispiel 1:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
    <t:SWITCH ID="ID_Switch">
        <SPAN systemLanguage="es">Spanisch</SPAN>
        <SPAN systemLanguage="pt">Portugiesisch</SPAN>
        <SPAN systemLanguage="en">Englisch</SPAN>
    </t:SWITCH>
    <t:AUDIO ID="ID_Audio"
        SRC="fun.wav"
        SYSTEMLANGUAGE="mi, en"
    >
    </ t:AUDIO>
</BODY>
</HTML>
```

Beispiel 2 für Verwendung von DirectMusic:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
    <t:AUDIO ID="ID_Audio"
        VOLUME="100"
        PLAYER=dmusic
        SRC="sample.sgt"
    >
</BODY>
</HTML>
```

Beispiel 3 für Abspielen einer Sounddatei per Timeline als Alternative zum audio Objekt

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<HEAD>
<BODY>
    <t:MEDIA
        ID="ID_Media"
        BEGIN="indefinite;"
        SRC="test.wmv"
        FILL="remove"
        onmediacomplete="Timer2.beginElement();"
    >
    </t:MEDIA>

    <BR>

    <BUTTON onclick="ID_Media.beginElement();">Start</BUTTON>
    <BUTTON onclick="ID_Media.endElement();">Stop</BUTTON>

    <BR>

    <B>Volume control:</B>&nbsp;

    <INPUT TYPE="radio" NAME="ID_RadioButtonGruppe"
        onpropertychange="ID_Media.mute='true';"
    >Mute

    <INPUT TYPE="radio" NAME="ID_RadioButtonGruppe"
        onpropertychange="ID_Media.mute='false'; ID_Media.volume=25;"
    >25% Volume
```



```

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe" CHECKED
        onpropertychange="ID_Media.mute='false'; ID_Media.volume=50;"
>50% Volume

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
        onpropertychange="ID_Media.mute='false'; ID_Media.volume=75;"
>75% Volume

< INPUT TYPE ="radio" NAME ="ID_RadioButtonGruppe"
        onpropertychange="ID_Media.mute='false'; ID_Media.volume=100;"
>100% Volume
</BODY>
</HTML>

```

Eigenschaften:

.abstract	Beschreibung einer Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei wird der Text unter ABSTRACT vom aktuellen Eintrag geliefert und nicht der Datei selbst
.accelerate	Beschleunigung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .decelerate
.author	Name des Autor der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von AUTHOR des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playlist: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.autoReverse	automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.Banner	Text laut BANNER des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playlist: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.BannerAbstract	Text laut BANNER und dort ABSTRACT des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playlist: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.BannerMoreInfo	Text laut BANNER und dort MOREINFO des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playlist: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.begin	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction
BOUNDARY	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.bufferingProgress	aktueller prozentualer Status des Pufferns eines Datenflusses (Data Stream) einer Media-Datei auf der Timeline Prozentwert des bisher erfolgten Pufferns nur für Media-Datei mit Datenfluss
.canPause	generelle Pausierungsmöglichkeit einer Media-Datei auf der Timeline
.canSeek	generelle Möglichkeit der Auswahl eines Zeitpunktes einer Media-Datei auf der Timeline Auswahl per Seek-Methoden .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame() nicht jeder Media-Typ unterstützt Seek
.clipBegin	Startzeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline Eigenschaft .canSeek muss true liefern



	nicht jeder Media-Typ unterstützt Clipping
.clipEnd	Endezeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline
	Eigenschaft .canSeek muss true liefern
	nicht jeder Media-Typ unterstützt Clipping
.copyright	Copyright der Media-Datei auf der Timeline
	bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von COPYRIGHT des aktiven Eintrages geliefert und nicht den der Datei.
	Track entspricht playItem Objekt
	Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.currentFrame	Nummer des aktuellen Frame einer Media-Datei auf der Timeline
	nicht alle Media-Typen unterstützen Frames
.decelerate	Verlangsamung des Elementes auf der Timeline
	hat keinen Einfluss auf die Dauer der Timeline
	auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur
	Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten
	wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
	siehe .accelerate
.downloadCurrent	Anzahl der bisher downgeloadeten Bytes beim Laden einer Media-Datei auf der Timeline
	(Laden des Daten-Stream in Form der Media-Datei)
	nur für Media-Datei mit Datenfluss
.downloadTotal	Anzahl der insgesamt downgeloadeten Bytes nach dem kompletten Laden einer Media-Datei auf der Timeline
	(Laden des Daten-Stream in Form der Media-Datei)
.dur	Dauer Objektaktivitäten laut Eigenschaft .timeAction
	alternativ: Eigenschaft .end
.end	Objektaktivitäten laut Eigenschaft .timeAction beenden
	ab IE 6.x
	alternativ: Eigenschaft .dur
.fill	Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber
	bevor die Timeline des Elternelementes endet
	ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet werden darf !!
.hasAudio	Media-Datei mit Audioinhalt auf der Timeline
	nicht alle Media-Typen können Audio beinhalten
	Stummschaltung oder Lautstärkeinstellungen sind dabei egal
.hasDownloadProgres	Start des Donloades einer Media-Datei auf der Timeline
.hasMedia	Objekt ist HTML-Media-Objekt
.hasPlayList	Verfügbarkeit der Behavior-Collection .style.time2.playList für Element auf der Timeline,
	also ob Element eine Playliste hat
.hasVisual	Media-Datei mit visuellem Inhalt auf der Timeline
	visuelle Daten werden auf dem Bildschirm sichtbar
	nicht alle Media-Typen können sichtbare Daten beinhalten
IMMEDIATEEND	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.isStreamed	Media-Datei mit Datenfluss (Data Stream) auf der Timeline
	nicht alle Media-Typen unterstützen Datenstrom
.latestMediaTime	Wartezeit für Start der Wiedergabe eines Media-Elementes auf der Timeline
	nicht für Media-Datei mit Datenfluss
LONGTRANSITION	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mediaDur	Länge der Media-Datei in Sekunden (Länge bei der Wiedergabe) auf der Timeline
	Ereignis onmediacomplete muss ausgelöst also die Media-Datei komplett geladen worden sein
.mediaHeight	aktuelle Höhe des Media-Elementes in Pixels auf der Timeline
	Standardgemäß wird die Elementhöhe aus der Höhenangabe in der Media-Datei verwendet
	(ansonsten STYLE-Angabe zu .style.height im Element kodieren)
.mediaWidth	aktuelle Breite des Media-Elementes in Pixels auf der Timeline
	Standardgemäß wird die Elementhöhe aus der Breitenangabe in der Media-Datei verwendet
	(ansonsten STYLE-Angabe zu .style.width im Element kodieren)
.mimeType	MIME-Typ (Multipurpose Internet Mail Extension-Typ) des Media-Elementes auf der Timeline
MODULATE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
MOTIFNAME	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mute	Audio aktiv oder aus (stumm) auf der Timeline
	wenn Eigenschaft für body Objekt kodiert, so alle Audio-Elemente im Body davon betroffen
	beeinflusst nicht die Systemlautstärke-Regelung von Windows oder anderen Soundmixern
	siehe Objekt bgsound
.player	Wiedergabeplayer einem Media-Element zuordnen für Wiedergabe auf der Timeline
	Player ist eine Softwarekomponente, die im System und im Browser installiert sein muss
	ab IE 6.x werden keine Plugins mehr unterstützt sondern nur noch ActiveX-Controls
	z.B. ist {6BF52A52-394A-11d3-B153-00C04F79FAA6} das ActiveX-Control für den
	Windows Media Player 7.1
.playerObject	Zeiger auf den Wiedergabeplayer, der das Media-Element auf der Timeline wiedergibt
	besonders zu verwenden, wenn Player über ActiveX-Control im Browser implementiert ist
	Zeiger ermöglicht, die Methoden und Eigenschaften des Players anzusprechen



	siehe Playerbeschreibung des Herstellers vom Player (im Falle von Microsoft sind die Methoden und Eigenschaften im jeweiligen SDK der Player-Version zu finden)
	Playerart laut .player
.rating	Rating der Media-Datei auf der Timeline
.repeatCount	aktuelle Nummer der Wiederholung bei Loop
.repeatDur	Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline verlangt kodierte Eigenschaft .dur oder .repeat darf nicht kodiert werden mit der Eigenschaft .repeatCount
.restart	generelle Restartmöglichkeit eines Elementes auf der Timeline ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!
SEGMENTTYPE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.speed	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
.src	Url einer Media-Datei auf der Timeline
.syncBehavior	Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes, sinnvoll vorallem bei Time-Container (Gruppen von Objekten) Es muss die Eigenschaft .syncmaster kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der Beschreibung zum jeweiligen Objekt zu entnehmen.
.syncMaster	siehe .syncTolerance und .syncmaster Synchronisierung der Animation des im Container liegenden Elementes auf Timeline (Container ist Master in der Synchronisierung), wobei ein Master nur genau ein zu synchronisierendes Element haben darf (Eineindeutigkeit). nur sinnvoll bei Zwangssynchronisierung (siehe .syncBehavior) ersetzt die Eigenschaft clockSource, da diese deprecated ist und nicht mehr verwendet werden darf !
.syncTolerance	siehe .syncTolerance und .syncBehavior zeitliche Toleranz in Sekunden für Zwangs-Synchronisation von Elementen auf der Timeline also wenn Eigenschaft .syncBehavior auf "locked" gesetzt ist sinnvoll vorallem bei Time-Container (Gruppen von Objekten) siehe .syncBehavior und .syncMaster
.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.timeAction	Aktion des Objektes in der Timeline Aktion nur ausführbar wenn Objekt aktiv ist UND Timeline aktiv ist
.timeContainer	Typ der Timeline des Objektes
.timeParent	Zeiger auf das Eltern-Timeline
.title	Titel der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von TITLE des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
TRANSITIONTYPE	Advanced Stream Redirector (ASX) -Datei: Playlisten-Datei
.type	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software MIME-Typ eines Media-Elementes auf der Timeline
.updateMode	Media-Datei zum Element laut Eigenschaft .src Art des Updates der Eigenschaften eines Elementes nach dem Start des Elementes auf der Timeline folgende Eigenschaften können geupdatet werden: .autoReverse .begin .dur .end .fill .repeatCount .repeatDur .speed
.volume	Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes für alle Audio-Elemente des body Objektes auch möglich Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus ! siehe Objekt bgsound

Methoden:

.activeTimeToParentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.activeTimeToSegmentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.beginElement()	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes.. per Eigenschaft .isActive das Element auf Aktivsein prüfen



<code>.beginElementAt()</code>	<p>Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren)</p> <p>also mit Wartezeit ab Beginn der Timeline des Elementes</p> <p>wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit</p> <p>wenn zusätzlich ein weiterer Start (oder mehrere Starts)</p> <p>mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert</p> <p>mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt</p> <p>also Neustart nach einem erfolgten Start</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p>
<code>.documentTimeToParentTime()</code>	Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
<code>.endElement()</code>	<p>aktives Element auf der Timeline stoppen</p> <p>identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline</p> <p>bzw. die Dauer (Duration) des Elementes abgearbeitet wurde</p> <p>Alle Kinder des Elementes erhalten Information über Start des Elternelementes</p> <p>und sind somit korrekt auf der Timeline des Elternelementes..</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p>
<code>.endElementAt()</code>	<p>aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes</p> <p>ab Beginn der Timeline stoppen</p> <p>sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes</p> <p>wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p>
<code>.parentTimeToActiveTime()</code>	Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren
<code>.parentTimeToDocumentTime()</code>	Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
<code>.pauseElement()</code>	<p>aktives Element auf Timeline pausieren lassen</p> <p>ersetzt Methode <code>pause()</code>, die deprecated ist und nicht mehr verwendet werden darf !</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p>
<code>.resetElement()</code>	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
<code>.resumeElement()</code>	<p>Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben</p> <p>falls Element nicht pausiert, passiert nichts</p> <p>ersetzt die Methode <code>resume()</code>, da sie deprecated ist und nicht mehr verwendet werden darf</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p>
<code>.seekActiveTime()</code>	<p>aktives Element animieren ab einem Zeitpunkt auf der Timeline</p> <p>wenn Element nicht aktiv, so Fehler erzeugt</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p> <p>alle Media-Typen für Element zulässig</p>
<code>.seekSegmentTime()</code>	<p>siehe Eigenschaft <code>.canSeek</code></p> <p>aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline</p> <p>ohne Wiederholung der Animation</p> <p>wenn Element nicht aktiv, so Fehler erzeugt</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p> <p>Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst</p> <p>nicht alle Media-Typen für Element zulässig</p> <p>wenn unzulässig, so kein Fehler</p>
<code>.seekTo()</code>	<p>siehe Eigenschaft <code>.canSeek</code></p> <p>aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline</p> <p>einschliesslich möglicher Wiederholungen der Animation</p> <p>wenn Element nicht aktiv, so Fehler erzeugt</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p> <p>nicht alle Media-Typen für Element zulässig</p> <p>wenn unzulässig, so kein Fehler</p>
<code>.seekToFrame()</code>	<p>siehe Eigenschaft <code>.canSeek</code></p> <p>Frame eines aktiven Elementes auf der Timeline anwählen</p> <p>wenn Element nicht aktiv, so Fehler erzeugt</p> <p>per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen</p> <p>nicht alle Media-Typen für Element zulässig</p> <p>wenn unzulässig, so kein Fehler</p>
<code>.segmentTimeToActiveTime()</code>	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
<code>.segmentTimeToSimpleTime()</code>	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
<code>.simpleTimeToSegmentTime()</code>	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren

Events:

<code>onbegin</code>	<p>erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird</p> <p>nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Element</p> <p>erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn</p>
----------------------	--



	Eigenschaft <code>.repeatCount > 1</code> ist für das Eltern-Element
onend	siehe <code>onend</code> und <code>onrepeat</code> erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut <code>.repeatCount</code> bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit des Kindelement ebenfalls endet nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code> siehe <code>onbegin</code> und <code>onrepeat</code>
onmediacomplete	erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde für Animation per Timeline Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline siehe <code>onmediaerror</code>
onmediaerror	erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält für Animation per Timeline Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline ersetzt das Ereignis <code>onmedialoadfailed</code> , das deprecated ist und nicht mehr verwendet werden darf! siehe <code>onmediacomplete</code>
onoutofsync	erzeugt wenn das Element seine Timeline verliert, also nicht mehr auf seiner Timeline animiert wird (nicht synchron zur Timeline des Elementes) sinnvoll nur bei Zwangssynchronisierung per <code>.syncBehavior</code> auf "locked": Eventhandler sollte neu synchronisieren z.B. Reset des Elementes durch <code>.resetElement()</code> siehe <code>onsyncrestored</code>
onpause	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei <code>body</code> Objekt
onrepeat	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe <code>onbegin</code>), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für <code>onrepeat</code> kodiert hat (kein Heraufreichen des Ereignisses <code>onrepeat</code> zum Elternelement) <code>.repeatCount</code> bzw. <code>.repeat</code> muss > 1 sein: <code>onrepeat</code> wird also <code>.repeatCount - 1</code> mal erzeugt Kind eines Elementes siehe <code>onend</code> , <code>onbegin</code> , <code>.repeatCount</code> und <code>.repeat</code>
onreset	erzeugt wenn Element zurückgesetzt wurde per Methode <code>.resetElement()</code> , also die Timeline des Elementes den aktuellen Wert laut <code>.begin</code> wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
onresume	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für <code>body</code> Objekt
onreverse	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also <code>.repeatCount > 0</code> ist) <code>.autoReverse</code> muss auf "true" stehen
onseek	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde: <code>.seekActiveTime()</code> <code>.seekSegmentTime()</code> <code>.seekTo()</code> <code>.seekToFrame()</code>
onsyncrestored	erzeugt wenn Element gerade mit seiner Timeline synchronisiert wird nach vorausgegangenem Abbruch der Synchronisation siehe <code>onoutofsync</code> sinnvoll nur bei Zwangssynchronisierung per <code>.syncBehavior</code> auf "locked"
ontrackchange	erzeugt wenn der Track einer Advanced Stream Redirector (ASX) Datei (*.asx) in der Playliste gewechselt wurde
onURLFlip	erzeugt wenn ein Skriptkommando, das in einer Advanced Streaming Format (ASF)-Datei (*.asf) liegen, ausgeführt wird Hinweis: <code>window.event.URL</code> Url laut einem Kommando aus einer Advanced Streaming Format (ASF)-Datei von Element auf der Timeline es muss Ereignis <code>onURLFlip</code> aufgetreten sein

4.3.2.2.4.3.39.23.9.3.6. *.style.time2.excl Behavior-Objekt des Internet Explorer*

Timerobjekt für exklusive Animation eines Elementes

Ein Kind in einem EXCL-Time-Container endet sofort, wenn ein anderes Kind mit der Animation startet. Es kann also immer nur 1 Kind animiert werden und das exklusiv gegenüber anderen Kindern.



Wenn ein Kind in einem EXCL-Time-Container ein bereits aktives Kind unterbrechen soll, wobei danach zum unterbrochenen Kind zurückgekehrt werden soll, so ist das Objekt priorityClass zu verwenden.

Syntax:

XML t:EXCL
Script excl

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:EXCL ID="ID_Excl"
ENDSYNC="ID_Div2"
>
<DIV ID="ID_Div1"
CLASS="time_line_klasse"
BEGIN="0"
DUR="2"
>
Zeile 1
</DIV>
<DIV ID="ID_Div2"
CLASS="time_line_klasse"
BEGIN="2"
DUR="2"
>
Zeile 2
</DIV>
<DIV ID="ID_Div3"
CLASS="time_line_klasse"
BEGIN="3"
DUR="2"
>
Zeile 3
</DIV>
</t:EXCL>
</BODY>
</HTML>
```

Eigenschaften:

.accelerate	Beschleunigung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .decelerate
.autoReverse	automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.decelerate	Verlangsamung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .accelerate
.dur	Dauer Objektaktivitäten laut Eigenschaft .timeAction alternativ: Eigenschaft .end
.endSync	Ende der Animation von Elementen in einem gemeinsamen Time-Container bei Ende der Timeline des Eltern-Time-Containers
.repeatDur	Verhalten des Eltern-Time-Containers bezüglich der Animation seiner Kind-Elemente Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline verlangt kodierte Eigenschaft .dur oder .repeat darf nicht kodiert werden mit der Eigenschaft .repeatCount
.syncBehavior	Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes, sinnvoll vorallem bei Time-Container (Gruppen von Objekten) Es muss die Eigenschaft .syncmaster kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der Beschreibung zum jeweiligen Objekt zu entnehmen.
.syncTolerance	siehe .syncTolerance und .syncmaster zeitliche Toleranz in Sekunden für Zwangs-Synchronisation von Elementen auf der Timeline also wenn Eigenschaft .syncBehavior auf "locked" gesetzt ist sinnvoll vorallem bei Time-Container (Gruppen von Objekten) siehe .syncBehavior und .syncMaster



Methoden:

<code>.activeTimeToParentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren
<code>.activeTimeToSegmentTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren
<code>.beginElement()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
<code>.documentTimeToParentTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
<code>.endElement()</code>	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
<code>.parentTimeToActiveTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren
<code>.parentTimeToDocumentTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
<code>.resetElement()</code>	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
<code>.seekTo()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.seekToFrame()</code>	Frame eines aktiven Elementes auf der Timeline anwählen wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.segmentTimeToActiveTime()</code>	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
<code>.segmentTimeToSimpleTime()</code>	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
<code>.simpleTimeToSegmentTime()</code>	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren

Events:

<code>onend</code>	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut <code>.repeatCount</code> bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit das Kindelement ebenfalls endet nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code> siehe <code>onbegin</code> und <code>onrepeat</code>
<code>onmediaerror</code>	erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält für Animation per Timeline Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline ersetzt das Ereignis <code>onmedialoadfailed</code> , das deprecated ist und nicht mehr verwendet werden darf ! siehe <code>onmediacomplete</code>
<code>onrepeat</code>	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe <code>onbegin</code>), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für <code>onrepeat</code> kodiert hat (kein Heraufreichen des Ereignisses <code>onrepeat</code> zum Elternelement) <code>.repeatCount</code> bzw. <code>.repeat</code> muss > 1 sein: <code>onrepeat</code> wird also <code>.repeatCount - 1</code> mal erzeugt Kind eines Elementes siehe <code>onend</code> , <code>onbegin</code> , <code>.repeatCount</code> und <code>.repeat</code>



onreset	erzeugt wenn Element zurückgesetzt wurde per Methode .resetElement(), also die Timeline des Elementes den aktuellen Wert laut .begin wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
onreverse	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist) .autoReverse muss auf "true" stehen
onseek	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde: .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame()

4.3.2.2.4.3.39.23.9.3.7. *.style.time2.img Behavior-Objekt des Internet Explorer*

Timerobjekt, das ein Bild (Image) animiert.

Syntax:

XML t:IMG
Script img Objekt des Behavior .style.time2

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:IMG ID="ID_Img"
SRC="test.gif"
STYLE="position:relative;top:25px;left:50px;height:100px;width:100px;"
onmediaerror="alert('Datei test.gif nicht ladbar !')"
onrepeat="alert('Aktuelle Wiederholung: ' + ID_Animation.currTimeState.repeatCount);"
>
</t:IMG>
<t:ANIMATION ID="ID_Animation"
TARGETELEMENT="ID_Img"
TO="0,400"
DUR="2"
BEGIN="0"
ACCELERATE="1"
AUTOREVERSE="true"
REPEATCOUNT="5"
>
</t:ANIMATION>
</BODY>
</HTML>
```

Eigenschaften:

.abstract	Beschreibung einer Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei wird der Text unter ABSTRACT vom aktuellen Eintrag geliefert und nicht der Datei selbst
.accelerate	Beschleunigung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
.author	siehe .decelerate Name des Autor der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von AUTHOR des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.autoReverse	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.begin	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction
BOUNDARY	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.bufferingProgress	aktueller prozentualer Status des Pufferns eines Datenflusses (Data Stream) einer Media-Datei auf der Timeline Prozentwert des bisher erfolgten Pufferns nur für Media-Datei mit Datenfluss
.canPause	generelle Pausierungsmöglichkeit einer Media-Datei auf der Timeline



.canSeek	generelle Möglichkeit der Auswahl eines Zeitpunktes einer Media-Datei auf der Timeline Auswahl per Seek-Methoden .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame()
.clipBegin	nicht jeder Media-Typ unterstützt Seek Startzeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline Eigenschaft .canSeek muss true liefern
.clipEnd	nicht jeder Media-Typ unterstützt Clipping Endezeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline Eigenschaft .canSeek muss true liefern
.copyright	nicht jeder Media-Typ unterstützt Clipping Copyright der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von COPYRIGHT des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.currentFrame	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei Nummer des aktuellen Frame einer Media-Datei auf der Timeline nicht alle Media-Typen unterstützen Frames
.decelerate	Verlangsamung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
.downloadCurrent	siehe .accelerate Anzahl der bisher downgeloadeten Bytes beim Laden einer Media-Datei auf der Timeline (Laden des Daten-Stream in Form der Media-Datei) nur für Media-Datei mit Datenfluss
.downloadTotal	Anzahl der insgesamt downgeloadeten Bytes nach dem kompletten Laden einer Media-Datei auf der Timeline (Laden des Daten-Stream in Form der Media-Datei)
.dur	Dauer Objektaktivitäten laut Eigenschaft .timeAction alternativ: Eigenschaft .end
.end	Objektaktivitäten laut Eigenschaft .timeAction beenden ab IE 6.x alternativ: Eigenschaft .dur
.fill	Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber bevor die Timeline des Elternelementes endet ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet werden darf !!
.hasAudio	Media-Datei mit Audioinhalt auf der Timeline nicht alle Media-Typen können Audio beinhalten Stummschaltung oder Lautstärkeinstellungen sind dabei egal
.hasDownloadProgres	Start des Donloades einer Media-Datei auf der Timeline
.hasMedia	Objekt ist HTML-Media-Objekt
.hasVisual	Media-Datei mit visuellem Inhalt auf der Timeline visuelle Daten werden auf dem Bildschirm sichtbar nicht alle Media-Typen können sichtbare Daten beinhalten
IMMEDIATEEND	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.isStreamed	Media-Datei mit Datenfluss (Data Stream) auf der Timeline nicht alle Media-Typen unterstützen Datenstrom
.latestMediaTime	Wartezeit für Start der Wiedergabe eines Media-Elementes auf der Timeline nicht für Media-Datei mit Datenfluss
LONGTRANSITION	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mediaDur	Länge der Media-Datei in Sekunden (Länge bei der Wiedergabe) auf der Timeline Ereignis onmediacomplete muss ausgelöst also die Media-Datei komplett geladen worden sein
.mediaHeight	aktuelle Höhe des Media-Elementes in Pixels auf der Timeline Standardgemäß wird die Elementhöhe aus der Höhenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.height im Element kodieren)
.mediaWidth	aktuelle Breite des Media-Elementes in Pixels auf der Timeline Standardgemäß wird die Elementhöhe aus der Breitenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.width im Element kodieren)
.mimeType	MIME-Typ (Multipurpose Internet Mail Extension-Typ) des Media-Elementes auf der Timeline
MODULATE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
MOTIFNAME	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mute	Audio aktiv oder aus (stumm) auf der Timeline wenn Eigenschaft für body Objekt kodiert, so alle Audio-Elemente im Body davon betroffen beeinflusst nicht die Systemlautstärke-Regelung von Windows oder anderen Soundmixern siehe Objekt bgsound
.player	Wiedergabeplayer einem Media-Element zuordnen für Wiedergabe auf der Timeline



	<p>Player ist eine Softwarekomponente, die im System und im Browser installiert sein muss ab IE 6.x werden keine Plugins mehr unterstützt sondern nur noch ActiveX-Controls z.B. ist {6BF52A52-394A-11d3-B153-00C04F79FAA6} das ActiveX-Control für den Windows Media Player 7.1</p>
.playerObject	<p>Zeiger auf den Wiedergabeplayer, der das Media-Element auf der Timeline wiedergibt besonders zu verwenden, wenn Player über ActiveX-Control im Browser implementiert ist Zeiger ermöglicht, die Methoden und Eigenschaften des Players anzusprechen siehe Playerbeschreibung des Herstellers vom Player (im Falle von Microsoft sind die Methoden und Eigenschaften im jeweiligen SDK der Player-Version zu finden)</p>
.rating	Playerart laut .player
.repeatCount	Rating der Media-Datei auf der Timeline
.repeatDur	aktuelle Nummer der Wiederholung bei Loop
	Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline
.restart	verlangt kodierte Eigenschaft .dur oder .repeat darf nicht kodiert werden mit der Eigenschaft .repeatCount generelle Restartmöglichkeit eines Elementes auf der Timeline
SEGMENTTYPE	ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!
.speed	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.src	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
.syncBehavior	Url einer Media-Datei auf der Timeline
	Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes, sinnvoll vorallem bei Time-Container (Gruppen von Objekten)
	Es muss die Eigenschaft .syncmaster kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der Beschreibung zum jeweiligen Objekt zu entnehmen.
.syncMaster	siehe .syncTolerance und .syncmaster Synchronisierung der Animation des im Container liegenden Elementes auf Timeline (Container ist Master in der Synchronisierung), wobei ein Master nur genau ein zu synchronisierendes Element haben darf (Eineindeutigkeit). nur sinnvoll bei Zwangssynchronisierung (siehe .syncBehavior) ersetzt die Eigenschaft clockSource, da diese deprecated ist und nicht mehr verwendet werden darf ! siehe .syncTolerance und .syncBehavior
.syncTolerance	zeitliche Toleranz in Sekunden für Zwangs-Synchronisation von Elementen auf der Timeline also wenn Eigenschaft .syncBehavior auf "locked" gesetzt ist sinnvoll vorallem bei Time-Container (Gruppen von Objekten) siehe .syncBehavior und .syncMaster
.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.timeAction	Aktion des Objektes in der Timeline
	Aktion nur ausführbar wenn Objekt aktiv ist UND Timeline aktiv ist
.timeContainer	Typ der Timeline des Objektes
.timeParent	Zeiger auf das Eltern-Timeline
.title	Titel der Media-Datei auf der Timeline
	bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von TITLE des aktiven Eintrages geliefert und nicht den der Datei.
	Track entspricht playItem Objekt
	Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
TRANSITIONTYPE	Advanced Stream Redirector (ASX) -Datei: Playlisten-Datei
.type	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
	MIME-Typ eines Media-Elementes auf der Timeline
.updateMode	Media-Datei zum Element laut Eigenschaft .src Art des Updates der Eigenschaften eines Elementes nach dem Start des Elementes auf der Timeline folgende Eigenschaften können geupdatet werden:
	.autoReverse
	.begin
	.dur
	.end
	.fill
	.repeatCount
	.repeatDur
	.speed
.volume	Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes für alle Audio-Elemente des body Objektes auch möglich Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus ! siehe Objekt bgsound

Methoden:

.activeTimeToParentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.activeTimeToSegmentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline



	des Segmentes konvertieren
.beginElement()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.beginElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start
.documentTimeToParentTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
.endElement()	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.endElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne
.parentTimeToActiveTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren
.parentTimeToDocumentTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
.pauseElement()	aktives Element auf Timeline pausieren lassen ersetzt Methode pause(), die deprecated ist und nicht mehr verwendet werden darf !
.resetElement()	per Eigenschaft .isActive das Element auf Aktivsein prüfen alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
.resumeElement()	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben falls Element nicht pausiert, passiert nichts ersetzt die Methode resume(), da sie deprecated ist und nicht mehr verwendet werden darf
.seekActiveTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element animieren ab einem Zeitpunkt auf der Timeline wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen
.seekSegmentTime()	alle Media-Typen für Element zulässig siehe Eigenschaft .canSeek aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline ohne Wiederholung der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft .canSeek
.seekTo()	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft .canSeek
.seekToFrame()	Frame eines aktiven Elementes auf der Timeline anwählen wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft .canSeek
.segmentTimeToActiveTime()	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
.segmentTimeToSimpleTime()	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
.simpleTimeToSegmentTime()	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der



Segment-Timeline des Elementes konvertieren

Events:

onbegin	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Element erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Eltern-Element siehe onend und onrepeat
onend	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut .repeatCount bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit das Kindelement ebenfalls endet nicht erzeugt wenn .fill mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode .endElement() und Eigenschaft .end siehe onbegin und onrepeat
onmediacomplete	erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde für Animation per Timeline Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline siehe onmediaerror
onmediaerror	erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält für Animation per Timeline Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline ersetzt das Ereignis onmedialoadfailed, das deprecated ist und nicht mehr verwendet werden darf ! siehe onmediacomplete
onoutofsync	erzeugt wenn das Element seine Timeline verliert, also nicht mehr auf seiner Timeline animiert wird (nicht synchron zur Timeline des Elementes) sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked": Eventhandler sollte neu synchronisieren z.B. Reset des Elementes durch .resetElement() siehe onsyncrestored
onpause	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt
onrepeat	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe onbegin), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für onrepeat kodiert hat (kein Heraufreichen des Ereignisses onrepeat zum Elternelement) .repeatCount bzw. .repeat muss > 1 sein: onrepeat wird also .repeatCount - 1 mal erzeugt Kind eines Elementes siehe onend, onbegin, .repeatCount und .repeat
onreset	erzeugt wenn Element zurückgesetzt wurde per Methode .resetElement(), also die Timeline des Elementes den aktuellen Wert laut .begin wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
onresume	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
onreverse	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist) .autoReverse muss auf "true" stehen
onseek	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde: .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame()
onsyncrestored	erzeugt wenn Element gerade mit seiner Timeline synchronisiert wird nach vorausgegangenem Abbruch der Synchronisation siehe onoutofsync sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked"

4.3.2.2.4.3.39.23.9.3.8.

.style.time2.media Behavior-Objekt des Internet Explorer

Timerobjekt zur Animation einer Media-Datei

Syntax:

XML t:MEDIA



Script media

Beispiel

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:MEDIA ID="ID_Media"
BEGIN="indefinite;"
SRC="test.wmv"
FILL="remove"
onmediacomplete="Timer2.beginElement();"
>
</t:MEDIA>

<BR>

<BUTTON onclick="ID_Media.beginElement();">Start</BUTTON>
<BUTTON onclick="ID_Media.endElement();">Stop</BUTTON>

<BR>

<B>Volume control:</B>&nbsp;

<INPUT TYPE="radio" NAME="ID_RadioButtonGruppe"
onpropertychange="ID_Media.mute='true';"
>Mute

<INPUT TYPE="radio" NAME="ID_RadioButtonGruppe"
onpropertychange="ID_Media.mute='false'; ID_Media.volume=25;"
>25% Volume

<INPUT TYPE="radio" NAME="ID_RadioButtonGruppe" CHECKED
onpropertychange="ID_Media.mute='false'; ID_Media.volume=50;"
>50% Volume

<INPUT TYPE="radio" NAME="ID_RadioButtonGruppe"
onpropertychange="ID_Media.mute='false'; ID_Media.volume=75;"
>75% Volume

<INPUT TYPE="radio" NAME="ID_RadioButtonGruppe"
onpropertychange="ID_Media.mute='false'; ID_Media.volume=100;"
>100% Volume
</BODY>
</HTML>
```

Eigenschaften:

.abstract	Beschreibung einer Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei wird der Text unter ABSTRACT vom aktuellen Eintrag geliefert und nicht der Datei selbst
.accelerate	Beschleunigung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .decelerate
.author	Name des Autor der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von AUTHOR des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.autoReverse	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.Banner	Text laut BANNER des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste:



	<p>Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.</p> <p>Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei</p>
.BannerAbstract	<p>Text laut BANNER und dort ABSTRACT des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playlist</p> <p>ab IE 6.x</p> <p>Track entspricht playItem Objekt</p> <p>Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:</p> <p>Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.</p>
.BannerMoreInfo	<p>Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei</p> <p>Text laut BANNER und dort MOREINFO des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playlist</p> <p>ab IE 6.x</p> <p>Track entspricht playItem Objekt</p> <p>Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:</p> <p>Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.</p>
.begin	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
BOUNDARY	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction
.bufferingProgress	<p>wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software</p> <p>aktueller prozentualer Status des Pufferns eines Datenflusses (Data Stream) einer Media-Datei auf der Timeline</p> <p>Prozentwert des bisher erfolgten Pufferns</p> <p>nur für Media-Datei mit Datenfluss</p>
.canPause	generelle Pausierungsmöglichkeit einer Media-Datei auf der Timeline
.canSeek	<p>generelle Möglichkeit der Auswahl eines Zeitpunktes einer Media-Datei auf der Timeline</p> <p>Auswahl per Seek-Methoden</p> <p>.seekActiveTime()</p> <p>.seekSegmentTime()</p> <p>.seekTo()</p> <p>.seekToFrame()</p>
.clipBegin	<p>nicht jeder Media-Typ unterstützt Seek</p> <p>Startzeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline</p> <p>Eigenschaft .canSeek muss true liefern</p>
.clipEnd	<p>nicht jeder Media-Typ unterstützt Clipping</p> <p>Endezeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline</p> <p>Eigenschaft .canSeek muss true liefern</p>
.copyright	<p>nicht jeder Media-Typ unterstützt Clipping</p> <p>Copyright der Media-Datei auf der Timeline</p> <p>bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von COPYRIGHT des aktiven Eintrages geliefert und nicht den der Datei.</p> <p>Track entspricht playItem Objekt</p> <p>Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:</p> <p>Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.</p>
.currentFrame	<p>Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei</p> <p>Nummer des aktuellen Frame einer Media-Datei auf der Timeline</p> <p>nicht alle Media-Typen unterstützen Frames</p>
.decelerate	<p>Verlangsamung des Elementes auf der Timeline</p> <p>hat keinen Einfluss auf die Dauer der Timeline</p> <p>auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur</p> <p>Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten</p> <p>wenn ja, so werden beide Attribute ignoriert, also nicht verwendet</p> <p>siehe .accelerate</p>
.downloadCurrent	<p>Anzahl der bisher downgeloadeten Bytes beim Laden einer Media-Datei auf der Timeline</p> <p>(Laden des Daten-Stream in Form der Media-Datei)</p> <p>nur für Media-Datei mit Datenfluss</p>
.downloadTotal	<p>Anzahl der insgesamt downgeloadeten Bytes nach dem kompletten Laden einer Media-Datei auf der Timeline</p> <p>(Laden des Daten-Stream in Form der Media-Datei)</p>
.dur	<p>Dauer Objektaktivitäten laut Eigenschaft .timeAction</p> <p>alternativ: Eigenschaft .end</p>
.end	<p>Objektaktivitäten laut Eigenschaft .timeAction beenden</p> <p>ab IE 6.x</p> <p>alternativ: Eigenschaft .dur</p>
.fill	<p>Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber bevor die Timeline des Elternelementes endet</p> <p>ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet werden darf !!</p>
.hasAudio	<p>Media-Datei mit Audioinhalt auf der Timeline</p> <p>nicht alle Media-Typen können Audio beinhalten</p> <p>Stummschaltung oder Lautstärkeinstellungen sind dabei egal</p>
.hasDownloadProgres	Start des Donloades einer Media-Datei auf der Timeline



.hasMedia	Objekt ist HTML-Media-Objekt
.hasPlaylist	Verfügbarkeit der Behavior-Collection .style.time2.playlist für Element auf der Timeline, also ob Element eine Playliste hat
.hasVisual	Media-Datei mit visuellem Inhalt auf der Timeline visuelle Daten werden auf dem Bildschirm sichtbar nicht alle Media-Typen können sichtbare Daten beinhalten
IMMEDIATEEND	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.isStreamed	Media-Datei mit Datenfluss (Data Stream) auf der Timeline nicht alle Media-Typen unterstützen Datenstrom
.latestMediaTime	Wartezeit für Start der Wiedergabe eines Media-Elementes auf der Timeline nicht für Media-Datei mit Datenfluss
LONGTRANSITION	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mediaDur	Länge der Media-Datei in Sekunden (Länge bei der Wiedergabe) auf der Timeline Ereignis onmediacomplete muss ausgelöst also die Media-Datei komplett geladen worden sein
.mediaHeight	aktuelle Höhe des Media-Elementes in Pixels auf der Timeline Standardgemäß wird die Elementhöhe aus der Höhenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.height im Element kodieren)
.mediaWidth	aktuelle Breite des Media-Elementes in Pixels auf der Timeline Standardgemäß wird die Elementhöhe aus der Breitenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.width im Element kodieren)
.mimeType	MIME-Typ (Multipurpose Internet Mail Extension-Typ) des Media-Elementes auf der Timeline
MODULATE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
MOTIFNAME	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mute	Audio aktiv oder aus (stumm) auf der Timeline wenn Eigenschaft für body Objekt kodiert, so alle Audio-Elemente im Body davon betroffen beeinflusst nicht die Systemlautstärke-Regelung von Windows oder anderen Soundmixern siehe Objekt bgsound
.player	Wiedergabeplayer einem Media-Element zuordnen für Wiedergabe auf der Timeline Player ist eine Softwarekomponente, die im System und im Browser installiert sein muss ab IE 6.x werden keine Plugins mehr unterstützt sondern nur noch ActiveX-Controls z.B. ist {6BF52A52-394A-11d3-B153-00C04F79FAA6} das ActiveX-Control für den Windows Media Player 7.1
.playerObject	Zeiger auf den Wiedergabeplayer, der das Media-Element auf der Timeline wiedergibt besonders zu verwenden, wenn Player über ActiveX-Control im Browser implementiert ist Zeiger ermöglicht, die Methoden und Eigenschaften des Players anzusprechen siehe Playerbeschreibung des Herstellers vom Player (im Falle von Microsoft sind die Methoden und Eigenschaften im jeweiligen SDK der Player-Version zu finden)
.rating	Playerart laut .player
.repeatCount	Rating der Media-Datei auf der Timeline
.repeatDur	aktuelle Nummer der Wiederholung bei Loop Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline verlangt kodierte Eigenschaft .dur oder .repeat darf nicht kodiert werden mit der Eigenschaft .repeatCount
.restart	generelle Restartmöglichkeit eines Elementes auf der Timeline ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!
SEGMENTTYPE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.speed	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
.src	Url einer Media-Datei auf der Timeline
.syncBehavior	Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes, sinnvoll vorallem bei Time-Container (Gruppen von Objekten) Es muss die Eigenschaft .syncmaster kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der Beschreibung zum jeweiligen Objekt zu entnehmen.
.syncMaster	siehe .syncTolerance und .syncmaster Synchronisierung der Animation des im Container liegenden Elementes auf Timeline (Container ist Master in der Synchronisierung), wobei ein Master nur genau ein zu synchronisierendes Element haben darf (Eineindeutigkeit). nur sinnvoll bei Zwangssynchronisierung (siehe .syncBehavior) ersetzt die Eigenschaft clockSource, da diese deprecated ist und nicht mehr verwendet werden darf ! siehe .syncTolerance und .syncBehavior
.syncTolerance	zeitliche Toleranz in Sekunden für Zwangs-Synchronisation von Elementen auf der Timeline also wenn Eigenschaft .syncBehavior auf "locked" gesetzt ist sinnvoll vorallem bei Time-Container (Gruppen von Objekten) siehe .syncBehavior und .syncMaster
.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.timeAction	Aktion des Objektes in der Timeline Aktion nur ausführbar wenn Objekt aktiv ist UND Timeline aktiv ist
.timeContainer	Typ der Timeline des Objektes
.timeParent	Zeiger auf das Eltern-Timeline
.title	Titel der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von TITLE des aktiven



	Eintrages geliefert und nicht den der Datei.
	Track entspricht playItem Objekt
	Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
TRANSITIONTYPE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.type	MIME-Typ eines Media-Elementes auf der Timeline
	Media-Datei zum Element laut Eigenschaft .src
.updateMode	Art des Updates der Eigenschaften eines Elementes nach dem Start des Elementes auf der Timeline
	folgende Eigenschaften können geupdatet werden:
	.autoReverse
	.begin
	.dur
	.end
	.fill
	.repeatCount
	.repeatDur
	.speed
.URL	Url aus einem Scriptkommando aus einer Advanced Streaming Format (ASF)-Datei
	es muss Event onURLFlip erzeugt worden sein
.volume	Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes
	für alle Audio-Elemente des body Objektes auch möglich
	Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus !
	siehe Objekt bgsound
Methoden:	
.activeTimeToParentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.activeTimeToSegmentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.beginElement()	Element auf der Timeline starten, also aktivieren
	identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline
	Alle Kinder des Elementes erhalten Information über Start des Elternelementes
	und sind somit korrekt auf der Timeline des Elternelementes..
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.beginElementAt()	Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren)
	also mit Wartezeit ab Beginn der Timeline des Elementes
	wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort,
	also ohne Wartezeit
	wenn zusätzlich ein weiterer Start (oder mehrere Starts)
	mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert
	mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt
	also Neustart nach einem erfolgten Start
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.documentTimeToParentTime()	Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
.endElement()	aktives Element auf der Timeline stoppen
	identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline
	bzw. die Dauer (Duration) des Elementes abgearbeitet wurde
	Alle Kinder des Elementes erhalten Information über Start des Elternelementes
	und sind somit korrekt auf der Timeline des Elternelementes..
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.endElementAt()	aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes
	ab Beginn der Timeline stoppen
	sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes
	wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort,
	also ohne Zeitspanne
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.nextTrack()	nächstes playItem Objekt aktivieren laut Behavior-Collection .style.time2.playList
	also Abspielen des Tracks starten
	bzw. nächste Wiederholung starten wenn .repeatCount > 0 und noch nicht
	alle Wiederholungen abgearbeitet wurden
	wenn letzter Track laut der Playliste aktiv ist und dann der nächste Track aktiviert werden soll,
	so wird nicht der erste Track abgespielt, sondern der aktive Track in der
	Wiedergabe gestoppt und danach keine weiteren Aktionen mehr
	Track entspricht playItem Objekt
	Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
	aktiver Track in der Liste: wird abgespielt
	Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten



<code>.parentTimeToActiveTime()</code>	Wert der Timeline der Eltern zum korrespondierenden Wert der aktives Timeline des Elementes konvertieren
<code>.parentTimeToDocumentTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
<code>.pauseElement()</code>	aktives Element auf Timeline pausieren lassen ersetzt Methode <code>pause()</code> , die deprecated ist und nicht mehr verwendet werden darf !
<code>.prevTrack()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen vorhergehendes <code>playItem</code> Objekt aktivieren laut Behavior-Collection <code>.style.time2.playlist</code> also Abspielen des Tracks starten aber nicht vorhergehende Wiederholung starten wenn <code>.repeatCount > 0</code> und noch nicht alle Wiederholungen abgearbeitet wurden wenn erster Track laut der Playliste aktiv ist und dann der vorhergehende Track aktiviert werden soll, so wird der erste Track nochmal abgespielt Track entspricht <code>playItem</code> Objekt Trackliste liegt in der Behavior-Collection <code>.style.time2.playlist</code> als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei aktiver Track in der Liste: wird abgespielt Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
<code>.resetElement()</code>	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben
<code>.resumeElement()</code>	falls Element nicht pausiert, passiert nichts ersetzt die Methode <code>resume()</code> , da sie deprecated ist und nicht mehr verwendet werden darf per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.seekActiveTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Timeline wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen alle Media-Typen für Element zulässig siehe Eigenschaft <code>.canSeek</code>
<code>.seekSegmentTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline ohne Wiederholung der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.seekTo()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.seekToFrame()</code>	Frame eines aktiven Elementes auf der Timeline anwählen wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.segmentTimeToActiveTime()</code>	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
<code>.segmentTimeToSimpleTime()</code>	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
<code>.simpleTimeToSegmentTime()</code>	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren

Events:

<code>onbegin</code>	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Element erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Eltern-Element siehe <code>onend</code> und <code>onrepeat</code>
<code>onend</code>	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut <code>.repeatCount</code> bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit des Kindelement ebenfalls endet nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code>



onmediacomplete	<p>siehe onbegin und onrepeat</p> <p>erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde für Animation per Timeline</p> <p>Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc.</p> <p>sinnvoll für Start des Elementes auf der Timeline</p> <p>siehe onmediaerror</p>
onmediaerror	<p>erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält für Animation per Timeline</p> <p>Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc.</p> <p>sinnvoll für Start des Elementes auf der Timeline</p> <p>ersetzt das Ereignis onmedialoadfailed, das deprecated ist und nicht mehr verwendet werden darf !</p>
onoutofsync	<p>erzeugt wenn das Element seine Timeline verliert, also nicht mehr auf seiner Timeline animiert wird (nicht synchron zur Timeline des Elementes)</p> <p>sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked": Eventhandler sollte neu synchronisieren z.B. Reset des Elementes durch .resetElement()</p>
onpause	<p>siehe onsyncrestored</p> <p>erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt</p>
onrepeat	<p>erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes</p> <p>nicht erzeugt beim Start des aktiven Elementes (siehe onbegin), also des ersten Durchlaufes, der keine Wiederholung ist</p> <p>nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für onrepeat kodiert hat (kein Heraufreichen des Ereignisses onrepeat zum Elternelement)</p> <p>.repeatCount bzw. .repeat muss > 1 sein:</p> <p>onrepeat wird also .repeatCount - 1 mal erzeugt</p> <p>Kind eines Elementes</p>
onreset	<p>siehe onend, onbegin, .repeatCount und .repeat</p> <p>erzeugt wenn Element zurückgesetzt wurde per Methode .resetElement(), also die Timeline des Elementes den aktuellen Wert laut .begin wieder erreicht und damit zurückgesetzt wurde</p> <p>also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes</p>
onresume	<p>erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt</p>
onreverse	<p>erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist)</p> <p>.autoReverse muss auf "true" stehen</p>
onseek	<p>erzeugt wenn eine Seek-Methode zum Element aktiviert wurde:</p> <p>.seekActiveTime()</p> <p>.seekSegmentTime()</p> <p>.seekTo()</p> <p>.seekToFrame()</p>
onsyncrestored	<p>erzeugt wenn Element gerade mit seiner Timeline synchronisiert wird nach vorausgegangenem Abbruch der Synchronisation</p> <p>siehe onoutofsync</p> <p>sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked"</p>
ontrackchange	<p>erzeugt wenn der Track einer Advanced Stream Redirector (ASX) Datei (*.asx) in der Playliste gewechselt wurde</p>
onURLFlip	<p>erzeugt wenn ein Scriptkommando, das in einer Advanced Streaming Format (ASF)-Datei (*.asf) liegen, ausgeführt wird</p> <p>Hinweis: window.event.URL Url laut einem Kommando aus einer Advanced Streaming Format (ASF)-Datei von Element auf der Timeline</p> <p>es muss Ereignis onURLFlip aufgetreten sein</p>

4.3.2.2.4.3.39.23.9.3.9. *.style.time2.par Behavior-Objekt des Internet Explorer*

Timer-Container für parallele Animation, wobei sich die Timeline der Kinder überlappen können, aber nicht müssen Standardgemäß beginnen alle Kinder mit Beginn der Eltern-Timeline.

Syntax:

XML t:PAR
Script par

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
```



```

</t:PAR ID="ID_Par"
  BEGIN="0"
  DUR="10"
  FILL="hold"
>
  <t:ANIMATECOLOR ID="ID_Animatecolor1"
    TARGETELEMENT="ID_Div1"
    ATTRIBUTENAME="background-color"
    VALUES="#0000FF;cyan"
    BEGIN="0"
    DUR="5"
    FILL="hold"
  >
  </t:ANIMATECOLOR>
  <t:ANIMATECOLOR ID="ID_Animatecolor2"
    TARGETELEMENT="ID_Div2"
    ATTRIBUTENAME="background-color"
    VALUES="cyan ;#0000FF"
    BEGIN="0"
    DUR="5"
    FILL="hold"
  >
</t:PAR>
<DIV ID="ID_Div1"
  CLASS="time_line_Klasse"
  STYLE="position: absolute; left: 68px; width: 279px; top: 260px; height: 217px;
    border: 1px solid black; background-color: green;
  "
  >
    animierter Div
</DIV>
<DIV ID="ID_Div2"
  CLASS="time_line_classe"
  STYLE="position: absolute; left: 112px; width: 188px; top: 318px; height: 98px;
    padding-left: 3; background-color: gray;
  "
  >
    animierter Div
</DIV>
</BODY>
</HTML>

```

Eigenschaften:

.accelerate	Beschleunigung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .decelerate
.autoReverse	automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.decelerate	Verlangsamung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .accelerate
.dur	Dauer Objektaktivitäten laut Eigenschaft .timeAction alternativ: Eigenschaft .end
.endSync	Ende der Animation von Elementen in einem gemeinsamen Time-Container bei Ende der Timeline des Eltern-Time-Containers
.fill	Verhalten des Eltern-Time-Containers bezüglich der Animation seiner Kind-Elemente Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber bevor die Timeline des Elternelementes endet ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet werden darf !!
.mediaDur	Länge der Media-Datei in Sekunden (Länge bei der Wiedergabe) auf der Timeline Ereignis onmediacomplete muss ausgelöst also die Media-Datei komplett geladen worden sein
.mediaHeight	aktuelle Höhe des Media-Elementes in Pixels auf der Timeline Standardgemäß wird die Elementhöhe aus der Höhenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.height im Element kodieren)
.mediaWidth	aktuelle Breite des Media-Elementes in Pixels auf der Timeline Standardgemäß wird die Elementhöhe aus der Breitenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.width im Element kodieren)



<code>.mute</code>	Audio aktiv oder aus (stumm) auf der Timeline wenn Eigenschaft für body Objekt kodiert, so alle Audio-Elemente im Body davon betroffen beeinflusst nicht die Systemlautstärke-Regelung von Windows oder anderen Soundmixern siehe Objekt <code>bgSound</code>
<code>.repeatCount</code>	aktuelle Nummer der Wiederholung bei Loop
<code>.repeatDur</code>	Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline verlangt kodierte Eigenschaft <code>.dur</code> oder <code>.repeat</code> darf nicht kodiert werden mit der Eigenschaft <code>.repeatCount</code>
<code>.restart</code>	generelle Restartmöglichkeit eines Elementes auf der Timeline ersetzt die Eigenschaft <code>eventRestart</code> , da diese deprecated ist und nicht mehr verwendet werden darf !!
<code>.speed</code>	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
<code>.syncBehavior</code>	Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes, sinnvoll vorallem bei Time-Container (Gruppen von Objekten) Es muss die Eigenschaft <code>.syncMaster</code> kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der Beschreibung zum jeweiligen Objekt zu entnehmen. siehe <code>.syncTolerance</code> und <code>.syncMaster</code>
<code>.syncTolerance</code>	zeitliche Toleranz in Sekunden für Zwangs-Synchronisation von Elementen auf der Timeline also wenn Eigenschaft <code>.syncBehavior</code> auf "locked" gesetzt ist sinnvoll vorallem bei Time-Container (Gruppen von Objekten) siehe <code>.syncBehavior</code> und <code>.syncMaster</code>
<code>.timeAction</code>	Aktion des Objektes in der Timeline Aktion nur ausführbar wenn Objekt aktiv ist UND Timeline aktiv ist
<code>.timeParent</code>	Zeiger auf das Eltern-Timeline
<code>.volume</code>	Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes für alle Audio-Elemente des body Objektes auch möglich Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus ! siehe Objekt <code>bgSound</code>

Methoden:

<code>.activeTimeToParentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.activeTimeToSegmentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.beginElement()</code>	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
<code>.beginElementAt()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.documentTimeToParentTime()</code>	Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
<code>.endElement()</code>	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
<code>.endElementAt()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne
<code>.nextTrack()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nächstes playItem Objekt aktivieren laut Behavior-Collection <code>.style.time2.playlist</code> also Abspielen des Tracks starten bzw. nächste Wiederholung starten wenn <code>.repeatCount > 0</code> und noch nicht alle Wiederholungen abgearbeitet wurden wenn letzter Track laut der Playliste aktiv ist und dann der nächste Track aktiviert werden soll, so wird nicht der erste Track abgespielt, sondern der aktive Track in der Wiedergabe gestoppt und danach keine weiteren Aktionen mehr Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection <code>.style.time2.playlist</code> als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei



	aktiver Track in der Liste: wird abgespielt
	Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten
.parentTimeToActiveTime()	Wert der Timeline der Eltern zum korrespondierenden Wert der aktives Timeline des Elementes konvertieren
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.parentTimeToDocumentTime()	Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
.pauseElement()	aktives Element auf Timeline pausieren lassen
	ersetzt Methode pause(), die deprecated ist und nicht mehr verwendet werden darf !
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.prevTrack()	vorhergehendes playItem Objekt aktivieren laut Behavior-Collection .style.time2.playlist
	also Abspielen des Tracks starten
	aber nicht vorhergehende Wiederholung starten wenn .repeatCount > 0 und noch nicht alle Wiederholungen abgearbeitet wurden
	wenn erster Track laut der Playliste aktiv ist und dann der vorhergehende Track aktiviert werden soll, so wird der erste Track nochmal abgespielt
	Track entspricht playItem Objekt
	Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
	aktiver Track in der Liste: wird abgespielt
	Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten
.resetElement()	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
.resumeElement()	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben
	falls Element nicht pausiert, passiert nichts
	ersetzt die Methode resume(), da sie deprecated ist und nicht mehr verwendet werden darf
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.seekActiveTime()	aktives Element animieren ab einem Zeitpunkt auf der Timeline
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	alle Media-Typen für Element zulässig
	siehe Eigenschaft .canSeek
.seekSegmentTime()	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline
	ohne Wiederholung der Animation
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.seekTo()	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline
	einschliesslich möglicher Wiederholungen der Animation
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.seekToFrame()	Frame eines aktiven Elementes auf der Timeline anwählen
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.segmentTimeToActiveTime()	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
.segmentTimeToSimpleTime()	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
.simpleTimeToSegmentTime()	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren

Events:

onbegin	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird
	nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Element
	erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Eltern-Element
	siehe onend und onrepeat
onend	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut .repeatCount
	bzw. wenn das aktive Element gestoppt wird
	erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat
	und somit des Kindelement ebenfalls endet
	nicht erzeugt wenn .fill mit Wert "freeze" oder "hold" für das Element kodiert wurde,



	es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode .endElement() und Eigenschaft .end
onmediacomplete	siehe onbegin und onrepeat erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde für Animation per Timeline Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline
onmediaerror	siehe onmediaerror erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält für Animation per Timeline Media-Element benötigt vor seiner Animation dem Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline ersetzt das Ereignis onmedialoadfailed, das deprecated ist und nicht mehr verwendet werden darf !
onoutofsync	siehe onmediacomplete erzeugt wenn das Element seine Timeline verliert, also nicht mehr auf seiner Timeline animiert wird (nicht synchron zur Timeline des Elementes) sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked": Eventhandler sollte neu synchronisieren z.B. Reset des Elementes durch .resetElement()
onpause	siehe onsyncrestored erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt
onrepeat	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe onbegin), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für onrepeat kodiert hat (kein Herausheben des Ereignisses onrepeat zum Elternelement) .repeatCount bzw. .repeat muss > 1 sein: onrepeat wird also .repeatCount - 1 mal erzeugt Kind eines Elementes
onreset	siehe onend, onbegin, .repeatCount und .repeat erzeugt wenn Element zurückgesetzt wurde per Methode .resetElement(), also die Timeline des Elementes den aktuellen Wert laut .begin wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
onresume	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
onreverse	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist) .autoReverse muss auf "true" stehen
onseek	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde: .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame()
onsyncrestored	erzeugt wenn Element gerade mit seiner Timeline synchronisiert wird nach vorausgegangenem Abbruch der Synchronisation siehe onoutofsync sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked"

4.3.2.2.4.3.39.23.9.3.10. *.style.time2.playItem Behavior-Objekt des Internet Explorer*

Timerobjekt, das einen Track (Playlisten-Eintrag) repräsentiert

aktiver Track in der Liste: wird gerade abgespielt

Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten

alle Tracks werden per Zeiger in der Behavior-Collection .style.time2.playlist gesammelt:

Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist,
also das Elternobjekt noch nicht auf seiner Timeline geendet ist.

Timerobjekt (Track) kann **nur über** die Behavior-Collection .style.time2.playlist referenziert werden

Playliste (Trackliste) stammt aus einer Advanced Stream Redirector (ASX)-Datei, die Playlisteneinträge per Script besitzt

Beispiel für Aufbau eines Eintrages in einer ASX-Datei:

```
<ASX Version="1.0" PreviewMode="No" >
<entry>
  <title>Testtitel</title>
  <author>Testautor</author>
  <copyright>Test 2002</copyright>
```



```

<abstract>WAV Datei</abstract>
<ref href=""></ref>
<banner href = "Testbild.gif" >
    <moreinfo href = "Test.doc"></moreinfo>
    <abstract>besuche www.test.de</abstract>
</banner>
</entry>
</ASX>

```

instanziert in der Behavior-Collection .style.time2.playlist als Sammlung der Zeiger aller Tracks (Playlisten-Einträge)
 Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist,
 also das Elternobjekt noch nicht auf seiner Timeline geendet ist.

Syntax:

ein beliebiger Track	zeiger_auf_timer_objekt.playlist.item().eigenschaft zeiger_auf_timer_objekt.playlist.item().methode
der aktive Track	zeiger_auf_timer_objekt.playlist.activeTrack.eigenschaft zeiger_auf_timer_objekt.playlist.activeTrack.methode
zeiger_auf_timer_objekt	laut ID-Attribut

Beispiel:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
<SCRIPT LANGUAGE="JScript">
    function ButtonUpdate()
    {
        if (ID_Media.currTimeState.isActive)
        {
            ID_Button1.disabled=true;
            ID_Button2.disabled=false;
            ID_Button3.disabled=false;
            ID_Button4.disabled=false;
        }
        else
        {
            ID_Button1.disabled=false;
            ID_Button2.disabled=true;
            ID_Button3.disabled=true;
            ID_Button4.disabled=true;
        }
    }

    function AnzeigeUpdate()
    {
        ID_Span1.innerText = "Titel: " + ID_Media.playlist.activeTrack.title;
        ID_Span3.innerText = "Autor: " + ID_Media.playlist.activeTrack.author;
        ID_Span3.innerText = "Abstract: " + ID_Media.playlist.activeTrack.abstract;
        ID_Span4.innerText = "Copyright: " + ID_Media.playlist.activeTrack.copyright;
        ID_Span5.innerText = "Filename: " + ID_Media.playlist.activeTrack.src;
        ID_Span6.innerText = "Banner: " + ID_Media.playlist.activeTrack.Banner;
        ID_Span7.innerText = "Banner Abstract: " + ID_Media.playlist.activeTrack.BannerAbstract;
        ID_Span8.innerText = "Banner MoreInfo: " + ID_Media.playlist.activeTrack.BannerMoreInfo;
    }

    function AnzeigeLoeschen()
    {
        ID_Span1.innerText = "Titel: ";
        ID_Span2.innerText = "Autor: ";
        ID_Span3.innerText = "Abstract: ";
        ID_Span4.innerText = "Copyright: ";
        ID_Span5.innerText = "Filename: ";
        ID_Span6.innerText = "Banner: ";
        ID_Span7.innerText = "Banner Abstract: ";
        ID_Span8.innerText = "Banner MoreInfo: ";
    }
    }
</SCRIPT>
</HEAD>

```



```

<BODY onload="ButtonUpdate()">

    <t:MEDIA          ID="ID_Media"
                      SRC="test.asx"
                      BEGIN="indefinite"
                      TIMEACTION="visibility"
                      onend="ButtonUpdate();"
                      ontrackchange="AnzeigeUpdate();"
                      onmediacomplete="ButtonUpdate();AnzeigeUpdate();"
    >
    </t:MEDIA>

    <SPAN ID="ID_Span1">Titel:</SPAN>
    <BR>
    <SPAN ID="ID_Span2">Autor:</SPAN>
    <BR>
    <SPAN ID="ID_Span3">Abstract:</SPAN>
    <BR>
    <SPAN ID="ID_Span4">Copyright:</SPAN>
    <BR>
    <SPAN ID="ID_Span5">Filename:</SPAN>
    <BR>
    <SPAN ID="ID_Span6">Banner:</SPAN>
    <BR>
    <SPAN ID="ID_Span7">Banner Abstract:</SPAN>
    <BR>
    <SPAN ID="ID_Span8">Banner MoreInfo:</SPAN>

    <BR>
    <BUTTON          ID="ID_Button1"
                    onclick="ID_Media.beginElement(); ButtonUpdate();"
    >
        Start
    </BUTTON>
    <BUTTON          ID="ID_Button2"
                    onclick="ID_Media.playList.nextTrack();"
    >
        naechster Track
    </BUTTON>
    <BUTTON          ID="ID_Button3"
                    onclick="ID_Media.playList.prevTrack();"
    >
        vorhergehender Track
    </BUTTON>
    <BUTTON          ID="ID_Button4"
                    onclick="ID_Media.endElement(); AnzeigeLoeschen();"
    >
        Stop
    </BUTTON>
</BODY>
</HTML>

```

Eigenschaften:

.abstract	Beschreibung einer Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei wird der Text unter ABSTRACT vom aktuellen Eintrag geliefert und nicht der Datei selbst
.author	Name des Autor der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von AUTHOR des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.begin	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.copyright	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction Copyright der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von COPYRIGHT des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.dur	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei Dauer Objektaktivitäten laut Eigenschaft .timeAction alternativ: Eigenschaft .end



.end	Objektaktivitäten laut Eigenschaft .timeAction beenden ab IE 6.x alternativ: Eigenschaft .dur
.hasMedia	Objekt ist HTML-Media-Objekt
.index	Index des Playlisten-Eintrages in der Behavior-Collection .style.time2.playList Diese Eigenschaft ist eigentlich nur sinnig für die Kodierung zum aktiven Track..
.rating	Rating der Media-Datei auf der Timeline
.src	Url einer Media-Datei auf der Timeline
.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.timeContainer	Typ der Timeline des Objektes
.title	Titel der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von TITLE des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei MIME-Typ eines Media-Elementes auf der Timeline Media-Datei zum Element laut Eigenschaft .src
.type	
Methoden:	
.setActive()	playItem Objekt als aktiven Track setzen und als solchen in der Behavior-Collection .style.time2.playList registrieren verändert .isActive Track entspricht playItem Objekt (Playlisten-Eintrag) Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei aktiver Track in der Liste: wird abgespielt Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten

4.3.2.2.4.3.39.23.9.3.11. *.style.time2.playList Behavior-Collection des Internet Explorer*

Feld aller playItem Objekte-Referenzen zum Element auf der Timeline

.style.time2.playItem Behavior-Objekt
repräsentiert einen Track (Playlisten-Eintrag)

aktiver Track in der Liste: wird gerade abgespielt

Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten

alle Tracks werden per Zeiger in der Behavior-Collection .style.time2.playList gesammelt:

Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist,
also das Elternobjekt noch nicht auf seiner Timeline geendet ist.

Timerobjekt (Track) kann **nur über** die Behavior-Collection .style.time2.playList referenziert werden

Playliste (Trackliste) stammt aus einer Advanced Stream Redirector (ASX)-Datei, die Playlisteneinträge per Script besitzt

Beispiel für Aufbau eines Eintrages in einer ASX-Datei:

```
<ASX Version="1.0" PreviewMode="No" >
<entry>
  <title>Testitel</title>
  <author>Testautor</author>
  <copyright>Test 2002</copyright>
  <abstract>WAV Datei</abstract>
  <ref href=""></ref>
  <banner href = "Testbild.gif" >
    <moreinfo href = "Test.doc"></moreinfo>
    <abstract>besuche www.test.de</abstract>
  </banner>
</entry>
</ASX>
```

instanziert in der Behavior-Collection .style.time2.playList als Sammlung der Zeiger aller Tracks (Playlisten-Einträge)

Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist,
also das Elternobjekt noch nicht auf seiner Timeline geendet ist.

Syntax:

```
[ var FeldZeiger = ] zeiger_auf_timer_objekt.playList
[ var FeldElementZeiger = ] zeiger_auf_timer_objekt.playList [Index ]
```

Index: Integer und ab 0
muss in [] kodiert sein



zeiger_auf_timer_objekt laut ID-Attribut

Hinweis: ein beliebiger Track

zeiger_auf_timer_objekt.playlist.item().eigenschaft
zeiger_auf_timer_objekt.playlist.item().methode

der aktive Track

zeiger_auf_timer_objekt.playlist.activeTrack.eigenschaft
zeiger_auf_timer_objekt.playlist.activeTrack.methode

zeiger_auf_timer_objekt laut ID-Attribut

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
<SCRIPT LANGUAGE="JScript">
    function ButtonUpdate()
    {
        if (ID_Media.currTimeState.isActive)
        {
            ID_Button1.disabled=true;
            ID_Button2.disabled=false;
            ID_Button3.disabled=false;
            ID_Button4.disabled=false;
        }
        else
        {
            ID_Button1.disabled=false;
            ID_Button2.disabled=true;
            ID_Button3.disabled=true;
            ID_Button4.disabled=true;
        }
    }

    function AnzeigeUpdate()
    {
        ID_Span1.innerText = "Titel: " + ID_Media.playlist.activeTrack.title;
        ID_Span3.innerText = "Autor: " + ID_Media.playlist.activeTrack.author;
        ID_Span3.innerText = "Abstract: " + ID_Media.playlist.activeTrack.abstract;
        ID_Span4.innerText = "Copyright: " + ID_Media.playlist.activeTrack.copyright;
        ID_Span5.innerText = "Filename: " + ID_Media.playlist.activeTrack.src;
        ID_Span6.innerText = "Banner: " + ID_Media.playlist.activeTrack.Banner;
        ID_Span7.innerText = "Banner Abstract: " + ID_Media.playlist.activeTrack.BannerAbstract;
        ID_Span8.innerText = "Banner MoreInfo: " + ID_Media.playlist.activeTrack.BannerMoreInfo;
    }

    function AnzeigeLoeschen()
    {
        ID_Span1.innerText = "Titel: ";
        ID_Span2.innerText = "Autor: ";
        ID_Span3.innerText = "Abstract: ";
        ID_Span4.innerText = "Copyright: ";
        ID_Span5.innerText = "Filename: ";
        ID_Span6.innerText = "Banner: ";
        ID_Span7.innerText = "Banner Abstract: ";
        ID_Span8.innerText = "Banner MoreInfo: ";
    }
</SCRIPT>
</HEAD>
<BODY onload="ButtonUpdate()">

    <t:MEDIA ID="ID_Media"
        SRC="test.asx"
        BEGIN="indefinite"
        TIMEACTION="visibility"
        onend="ButtonUpdate();"
        ontrackchange="AnzeigeUpdate();"
        onmediacomplete="ButtonUpdate();AnzeigeUpdate();"

    >
```



```

</t:MEDIA>

<SPAN ID="ID_Span1">Titel:</SPAN>
<BR>
<SPAN ID="ID_Span2">Autor:</SPAN>
<BR>
<SPAN ID="ID_Span3">Abstract:</SPAN>
<BR>
<SPAN ID="ID_Span4">Copyright:</SPAN>
<BR>
<SPAN ID="ID_Span5">Filename:</SPAN>
<BR>
<SPAN ID="ID_Span6">Banner:</SPAN>
<BR>
<SPAN ID="ID_Span7">Banner Abstract:</SPAN>
<BR>
<SPAN ID="ID_Span8">Banner MoreInfo:</SPAN>

<BR>
<BUTTON ID="ID_Button1"
        onclick="ID_Media.beginElement(); ButtonUpdate();"
>
    Start
</BUTTON>
<BUTTON ID="ID_Button2"
        onclick="ID_Media.playList.nextTrack();"
>
    naechster Track
</BUTTON>
<BUTTON ID="ID_Button3"
        onclick="ID_Media.playList.prevTrack();"
>
    vorhergehender Track
</BUTTON>
<BUTTON ID="ID_Button4"
        onclick="ID_Media.endElement(); AnzeigeLoeschen();"
>
    Stop
</BUTTON>
</BODY>
</HTML>

```

Beispiel 2:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
    function Anzeige()
    {alert('hasPlayList: ' + ID_Media.hasPlayList); }
</SCRIPT>
</HEAD>
<BODY>
    <t:MEDIA ID="ID_Media"
            SRC="test.wmv"
            BEGIN="0"
            FILL="remove"
            onmediacomplete="Anzeige();"
    >
    </t:MEDIA>
</BODY>
</HTML>

```

Eigenschaften:

.activeTrack Referenz auf aktives Objekt playItem, das gerade in der Playliste aktiv ist
 Achtung: Playliste muss aktiv sein, sonst wird Fehler erzeugt
 Aktivstatus prüfen per Eigenschaft .isActive
 Track entspricht playItem Objekt
 Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste:
 Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
 Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
 aktiver Track in der Liste: wird abgespielt



.Banner	<p>Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten</p> <p>Text laut BANNER des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playlist</p> <p>ab IE 6.x</p> <p>Track entspricht playItem Objekt</p> <p>Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:</p> <p>Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.</p>
.BannerAbstract	<p>Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei</p> <p>Text laut BANNER und dort ABSTRACT des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playlist</p> <p>ab IE 6.x</p> <p>Track entspricht playItem Objekt</p> <p>Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:</p> <p>Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.</p>
.BannerMoreInfo	<p>Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei</p> <p>Text laut BANNER und dort MOREINFO des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playlist</p> <p>ab IE 6.x</p> <p>Track entspricht playItem Objekt</p> <p>Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:</p> <p>Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.</p>
.dur	<p>Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei</p> <p>Dauer Objektaktivitäten laut Eigenschaft .timeAction</p> <p>alternativ: Eigenschaft .end</p>
.length	Anzahl der Feldelemente also Feldlänge z.B. bei Collection
Methoden:	
.item()	<p>Referenz auf Feldelement anhand des Integer-Indexes oder des Attributnamen (analog zu ID oder NAME-Attribut) liefern</p> <p>außer bei Formular mit <INPUT TYPE=image ...> da dafür die children-Collection verwendet werden muss !!!</p>
.nextTrack()	<p>nächstes playItem Objekt aktivieren laut Behavior-Collection .style.time2.playlist</p> <p>also Abspielen des Tracks starten</p> <p>bzw. nächste Wiederholung starten wenn .repeatCount > 0 und noch nicht alle Wiederholungen abgearbeitet wurden</p> <p>wenn letzter Track laut der Playliste aktiv ist und dann der nächste Track aktiviert werden soll, so wird nicht der erste Track abgespielt, sondern der aktive Track in der Wiedergabe gestoppt und danach keine weiteren Aktionen mehr</p> <p>Track entspricht playItem Objekt</p> <p>Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:</p> <p>Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.</p>
.prevTrack()	<p>Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei</p> <p>aktiver Track in der Liste: wird abgespielt</p> <p>Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten</p> <p>vorhergehendes playItem Objekt aktivieren laut Behavior-Collection .style.time2.playlist</p> <p>also Abspielen des Tracks starten</p> <p>aber nicht vorhergehende Wiederholung starten wenn .repeatCount > 0 und noch nicht alle Wiederholungen abgearbeitet wurden</p> <p>wenn erster Track laut der Playliste aktiv ist und dann der vorhergehende Track aktiviert werden soll, so wird der erste Track nochmal abgespielt</p> <p>Track entspricht playItem Objekt</p> <p>Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:</p> <p>Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.</p>
4.3.2.2.4.3.39.23.9.3.12.	<i>.style.time2.priorityClass Behavior-Objekt des Internet Explorer</i>
Timerobjekt	<p>als Kind vom Time-Container t:EXCL</p> <p>definiert das Animations-Verhalten mehrerer Timerobjekte t:PRIORITYCLASS</p> <p>oder aller Elemente innerhalb eines gemeinsamen t:PRIORITYCLASS</p> <p>im Time-Container t:EXCL</p>
Time-Container t:EXCL	<p>jedes Kind ein t:PRIORITYCLASS</p> <p>alle Kinder sequentiell kodieren: Verschachtelung von t:PRIORITYCLASS nicht zulässig ist</p> <p>Kodierungsfolge ist die absteigende Prioritätsfolge für die Animation.</p> <p>Das erste kodierte t:PRIORITYCLASS hat die höchste Priorität</p> <p>Das letzte kodierte t:PRIORITYCLASS hat die niedrigste Priorität.</p>



Im ersten kodierten t:PRIORITYCLASS darf kein Klassen-Attribut HIGHER kodiert sein, dafür aber LOWER möglich. Soll LOWER kodiert werden, so muss es mindestens 1 Nachfolger-Kind geben.

Im letzten kodierten t:PRIORITYCLASS darf kein Klassen-Attribut LOWER kodiert sein, dafür aber HIGHER möglich. Soll HIGHER kodiert werden, so muss es mindestens 1 Vorgänger-Kind geben.

Es gibt **genau ein** t:PRIORITYCLASS für die Kodierung des Klassen-Attributes PEERS.

Syntax:

XML <t:PRIORITYCLASS klassen_attribut = kette >
Script priorityClass

klassen_attribut Gross-Klein egal

HIGHER Abbruch bzw. Pausierung der aktiven Animation durch Start der Animation

eines höher priorisierten Kindes

Kette String

Es gilt folgende Ausgangssituation:

Kind X hat kodiertes HIGHER animiert bereits

Kind Y hat höhere Priorität als Kind X animiert nicht

"pause" Default
wenn Kind Y startet, pausiert
solange, bis Kind Y fertig ist

"stop" wenn Kind Y startet, bricht Kind X ab

kein Abbruch von **bereits** pausierenden Kindern

LOWER Abbruch bzw. Pausierung der aktiven Animation durch Start der Animation eines niedriger priorisierten Kindes

Kette String

Es gilt folgende Ausgangssituation:

Kind X hat niedrigere Priorität als Kind Y animiert nicht

Kind Y hat LOWER kodiert animiert bereits

"defer" Default
wenn Kind X startet, pausiert
es sofort und wartet, bis Kind Y fertig ist

ist

"never" Kind X startet nie

PEERS Abbruch bzw. Pausierung der aktiven Animation durch Start der Animation eines Kindes

Kette String

Es gilt folgende Ausgangssituation:

Kind X animiert nicht
Kind Y animiert bereits

"pause" wenn Kind X startet, pausiert Kind Y
solange, bis Kind X fertig ist

"stop" Default
wenn Kind X startet, bricht Kind Y ab

kein Abbruch von **bereits** pausierenden Kindern



"defer"	wenn Kind X startet, dann pausiert es sofort und solange, bis Kind Y fertig ist
"never"	Kind X startet nie
kein	Abbruch von bereits pausierenden Kindern

Beispiel 1:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:EXCL ID="ID_Excl"
BEGIN="0;indefinite;"
>
<t:PRIORITYCLASS ID="ID_Prior1">
<SPAN ID="ID_Span1"
CLASS="time_line_Klasse"
BEGIN="5"
DUR="5"
>
Test1
</SPAN>
</t:PRIORITYCLASS>
<t:PRIORITYCLASS ID="ID_Prior2" HIGHER="stop">
<SPAN ID="ID_Span2"
CLASS="time_line_Klasse"
BEGIN="0"
DUR="10"
>
Test2
</SPAN>
</t:PRIORITYCLASS>
</t:EXCL>
<BR>
<BUTTON ID="ID_Button" onclick="ID_Excl.beginElement();">
</BUTTON>
</BODY>
</HTML>
```

Beispiel 2:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:EXCL ID="ID_Excl"
BEGIN="0;indefinite;"
>
<t:PRIORITYCLASS ID="ID_Prior1">
<SPAN ID="ID_Span1"
CLASS="time_line_Klasse"
BEGIN="5"
DUR="5"
>
Test1
</SPAN>
</t:PRIORITYCLASS>
<t:PRIORITYCLASS ID="ID_Prior2" LOWER="never">
<SPAN ID="ID_Span2"
CLASS="time_line_Klasse"
BEGIN="0"
DUR="10"
>
```



```

        >
            Test2
        </SPAN>
    </t:PRIORITYCLASS>
</t:EXCL>
<BR>
<BUTTON ID="ID_Button" onclick="ID_Excl.beginElement();">
</BUTTON>
</BODY>
</HTML>

```

Beispiel 3:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
    <t:EXCL ID="ID_Excl"
        BEGIN="0;indefinite;"
    >
        <t:PRIORITYCLASS ID="ID_Prior" PEERS="pause">
            <SPAN ID="ID_Span1"
                CLASS="time_line_Klasse"
                BEGIN="0"
                DUR="10"
            >
                Test1
            </SPAN>
            <SPAN ID="ID_Span2"
                CLASS="time_line_Klasse"
                BEGIN="5"
                DUR="3"
            >
                Test2
            </SPAN>
        </t:PRIORITYCLASS>
    </t:EXCL>
    <BR>
    <BUTTON ID="ID_Button" onclick="ID_Excl.beginElement();">
    </BUTTON>
</BODY>
</HTML>

```

Eigenschaften:

.higher	<p>Abbruch bzw. Pausierung der aktiven Animation durch Start der Animation eines höher priorisierten Kindes</p> <p>Time-Container ist Behavior-Objektes .style.time2.excl</p> <p>Folge der in HTML kodierten Objekte t:PRIORITYCLASS entspricht der absteigenden Folge der Prioritäten der Animation.</p> <p>Jedes Kind ist ein t:PRIORITYCLASS</p> <p>darf kein weiteres Behavior-Objekt .style.time2.priorityClass besitzen</p> <p>wenn mehrere pausierende Kinder existieren:</p> <p>Das zuletzt pausierte Kind startet zuerst.</p> <p>Das zuerst pausierte Kind startet zuletzt.</p>
.lower	<p>Abbruch bzw. Pausierung der aktiven Animation durch Start der Animation eines niedriger priorisierten Kindes</p> <p>Time-Container ist Behavior-Objektes .style.time2.excl</p> <p>Folge der in HTML kodierten Objekte t:PRIORITYCLASS entspricht der absteigenden Folge der Prioritäten der Animation.</p> <p>Jedes Kind ist ein t:PRIORITYCLASS</p> <p>darf kein weiteres Behavior-Objekt .style.time2.priorityClass besitzen</p> <p>wenn mehrere pausierende Kinder existieren:</p> <p>Das zuletzt pausierte Kind startet zuerst.</p> <p>Das zuerst pausierte Kind startet zuletzt.</p>
.peers	<p>Abbruch bzw. Pausierung der aktiven Animation durch Start der Animation eines Kindes</p> <p>Time-Container ist Behavior-Objektes .style.time2.excl</p> <p>Es gibt genau ein t:PRIORITYCLASS, in dem kein weiteres Behavior-Objekt .style.time2.priorityClass kodiert sein darf.</p> <p>wenn mehrere pausierende Kinder existieren:</p> <p>Das zuletzt pausierte Kind startet zuerst.</p> <p>Das zuerst pausierte Kind startet zuletzt.</p> <p>siehe Objekt currTimeState und Behavior .style.time2</p>



Methoden:

keine

4.3.2.2.4.3.39.23.9.3.13. *.style.time2.ref Behavior-Objekt des Internet Explorer*

Timerobjekt für Referenz innerhalb des Dokumentes

Syntax:

XML	t:REF
Script	ref

Eigenschaften:

.abstract	Beschreibung einer Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei wird der Text unter ABSTRACT vom aktuellen Eintrag geliefert und nicht der Datei selbst
.accelerate	Beschleunigung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .decelerate
.author	Name des Autor der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von AUTHOR des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.autoReverse	automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.Banner	Text laut BANNER des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.BannerAbstract	Text laut BANNER und dort ABSTRACT des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.BannerMoreInfo	Text laut BANNER und dort MOREINFO des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.begin	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction
BOUNDARY	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.bufferingProgress	aktueller prozentualer Status des Pufferns eines Datenflusses (Data Stream) einer Media-Datei auf der Timeline Prozentwert des bisher erfolgten Pufferns nur für Media-Datei mit Datenfluss
.canPause	generelle Pausierungsmöglichkeit einer Media-Datei auf der Timeline
.canSeek	generelle Möglichkeit der Auswahl eines Zeitpunktes einer Media-Datei auf der Timeline Auswahl per Seek-Methoden .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame() nicht jeder Media-Typ unterstützt Seek
.clipBegin	Startzeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline Eigenschaft .canSeek muss true liefern nicht jeder Media-Typ unterstützt Clipping
.clipEnd	Endezeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline Eigenschaft .canSeek muss true liefern nicht jeder Media-Typ unterstützt Clipping
.copyright	Copyright der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von COPYRIGHT des aktiven Eintrages geliefert und nicht den der Datei.



	Track entspricht playItem Objekt
	Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.currentFrame	Nummer des aktuellen Frame einer Media-Datei auf der Timeline
	nicht alle Media-Typen unterstützen Frames
.decelerate	Verlangsamung des Elementes auf der Timeline
	hat keinen Einfluss auf die Dauer der Timeline
	auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur
	Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten
	wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
	siehe .accelerate
.downloadCurrent	Anzahl der bisher downgeloadeten Bytes beim Laden einer Media-Datei auf der Timeline
	(Laden des Daten-Stream in Form der Media-Datei)
	nur für Media-Datei mit Datenfluss
.downloadTotal	Anzahl der insgesamt downgeloadeten Bytes nach dem kompletten Laden einer Media-Datei
	auf der Timeline
	(Laden des Daten-Stream in Form der Media-Datei)
.dur	Dauer Objektaktivitäten laut Eigenschaft .timeAction
	alternativ: Eigenschaft .end
.end	Objektaktivitäten laut Eigenschaft .timeAction beenden
	ab IE 6.x
	alternativ: Eigenschaft .dur
.fill	Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber
	bevor die Timeline des Elternelementes endet
	ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet werden darf !!
.hasAudio	Media-Datei mit Audioinhalt auf der Timeline
	nicht alle Media-Typen können Audio beinhalten
	Stummschaltung oder Lautstärkeinstellungen sind dabei egal
.hasDownloadProgres	Start des Donloades einer Media-Datei auf der Timeline
.hasMedia	Objekt ist HTML-Media-Objekt
.hasPlaylist	Verfügbarkeit der Behavior-Collection .style.time2.playlist für Element auf der Timeline,
	also ob Element eine Playliste hat
.hasVisual	Media-Datei mit visuellem Inhalt auf der Timeline
	visuelle Daten werden auf dem Bildschirm sichtbar
	nicht alle Media-Typen können sichtbare Daten beinhalten
IMMEDIATEEND	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.isStreamed	Media-Datei mit Datenfluss (Data Stream) auf der Timeline
	nicht alle Media-Typen unterstützen Datenstrom
.latestMediaTime	Wartezeit für Start der Wiedergabe eines Media-Elementes auf der Timeline
	nicht für Media-Datei mit Datenfluss
LONGTRANSITION	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mediaDur	Länge der Media-Datei in Sekunden (Länge bei der Wiedergabe) auf der Timeline
	Ereignis onmediacomplete muss ausgelöst also die Media-Datei komplett geladen worden sein
.mediaHeight	aktuelle Höhe des Media-Elementes in Pixels auf der Timeline
	Standardgemäß wird die Elementhöhe aus der Höhenangabe in der Media-Datei verwendet
	(ansonsten STYLE-Angabe zu .style.height im Element kodieren)
.mediaWidth	aktuelle Breite des Media-Elementes in Pixels auf der Timeline
	Standardgemäß wird die Elementhöhe aus der Breitenangabe in der Media-Datei verwendet
	(ansonsten STYLE-Angabe zu .style.width im Element kodieren)
.mimeType	MIME-Typ (Multipurpose Internet Mail Extension-Typ) des Media-Elementes auf der Timeline
MODULATE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
MOTIFNAME	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mute	Audio aktiv oder aus (stumm) auf der Timeline
	wenn Eigenschaft für body Objekt kodiert, so alle Audio-Elemente im Body davon betroffen
	beeinflusst nicht die Systemlautstärke-Regelung von Windows oder anderen Soundmixern
	siehe Objekt bgsound
.player	Wiedergabeplayer einem Media-Element zuordnen für Wiedergabe auf der Timeline
	Player ist eine Softwarekomponente, die im System und im Browser installiert sein muss
	ab IE 6.x werden keine Plugins mehr unterstützt sondern nur noch ActiveX-Controls
	z.B. ist {22d6f312-b0f6-11d0-94ab-0080c74c7e95} das ActiveX-Control für den
	Windows Media-Player im Internet Explorer.
.playerObject	Zeiger auf den Wiedergabeplayer, der das Media-Element auf der Timeline wiedergibt
	besonders zu verwenden, wenn Player über ActiveX-Control im Browser implementiert ist
	Zeiger ermöglicht, die Methoden und Eigenschaften des Players anzusprechen
	siehe Playerbeschreibung des Herstellers vom Player
	(im Falle von Microsoft sind die Methoden und Eigenschaften
	im jeweiligen SDK der Player-Version zu finden)
	Playerart laut .player
.rating	Rating der Media-Datei auf der Timeline
.repeatCount	aktuelle Nummer der Wiederholung bei Loop
.repeatDur	Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline



	verlangt kodierte Eigenschaft .dur oder .repeat darf nicht kodiert werden mit der Eigenschaft .repeatCount
.restart	generelle Restartmöglichkeit eines Elementes auf der Timeline ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!
SEGMENTTYPE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.speed	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
.src	Url einer Media-Datei auf der Timeline
.syncBehavior	Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes, sinnvoll vorallem bei Time-Container (Gruppen von Objekten) Es muss die Eigenschaft .syncmaster kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der Beschreibung zum jeweiligen Objekt zu entnehmen.
.syncMaster	siehe .syncTolerance und .syncmaster Synchronisierung der Animation des im Container liegenden Elementes auf Timeline (Container ist Master in der Synchronisierung), wobei ein Master nur genau ein zu synchronisierendes Element haben darf (Eineindeutigkeit). nur sinnvoll bei Zwangssynchronisierung (siehe .syncBehavior) ersetzt die Eigenschaft clockSource, da diese deprecated ist und nicht mehr verwendet werden darf ! siehe .syncTolerance und .syncBehavior
.syncTolerance	zeitliche Toleranz für Zwangs-Synchronisation von Elementen auf der Timeline also wenn Eigenschaft .syncBehavior auf "locked" gesetzt ist sinnvoll vorallem bei Time-Container (Gruppen von Objekten) siehe .syncBehavior und .syncMaster
.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.timeAction	Aktion des Objektes in der Timeline Aktion nur ausführbar wenn Objekt aktiv ist UND Timeline aktiv ist
.timeContainer	Typ der Timeline des Objektes
.timeParent	Zeiger auf das Eltern-Timeline
.title	Titel der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von TITLE des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
TRANSITIONTYPE	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.type	MIME-Typ eines Media-Elementes auf der Timeline Media-Datei zum Element laut Eigenschaft .src
.updateMode	Art des Updates der Eigenschaften eines Elementes nach dem Start des Elementes auf der Timeline folgende Eigenschaften können geupdatet werden: .autoReverse .begin .dur .end .fill .repeatCount .repeatDur .speed
.volume	Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes für alle Audio-Elemente des body Objektes auch möglich Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus ! siehe Objekt bgsound

Methoden:

.activeTimeToParentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.activeTimeToSegmentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.beginElement()	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.beginElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt



	also Neustart nach einem erfolgten Start
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.documentTimeToParentTime()	Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
.endElement()	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.endElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne
.nextTrack()	per Eigenschaft .isActive das Element auf Aktivsein prüfen nächstes playItem Objekt aktivieren laut Behavior-Collection .style.time2.playlist also Abspielen des Tracks starten bzw. nächste Wiederholung starten wenn .repeatCount > 0 und noch nicht alle Wiederholungen abgearbeitet wurden wenn letzter Track laut der Playliste aktiv ist und dann der nächste Track aktiviert werden soll, so wird nicht der erste Track abgespielt, sondern der aktive Track in der Wiedergabe gestoppt und danach keine weiteren Aktionen mehr Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei aktiver Track in der Liste: wird abgespielt Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten
.parentTimeToActiveTime()	Wert der Timeline der Eltern zum korrespondierenden Wert der aktives Timeline des Elementes konvertieren
.parentTimeToDocumentTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
.pauseElement()	aktives Element auf Timeline pausieren lassen ersetzt Methode pause(), die deprecated ist und nicht mehr verwendet werden darf !
.prevTrack()	per Eigenschaft .isActive das Element auf Aktivsein prüfen vorhergehendes playItem Objekt aktivieren laut Behavior-Collection .style.time2.playlist also Abspielen des Tracks starten aber nicht vorhergehende Wiederholung starten wenn .repeatCount > 0 und noch nicht alle Wiederholungen abgearbeitet wurden wenn erster Track laut der Playliste aktiv ist und dann der vorhergehende Track aktiviert werden soll, so wird der erste Track nochmal abgespielt Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei aktiver Track in der Liste: wird abgespielt Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten
.resetElement()	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
.resumeElement()	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben falls Element nicht pausiert, passiert nichts ersetzt die Methode resume(), da sie deprecated ist und nicht mehr verwendet werden darf
.seekActiveTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element animieren ab einem Zeitpunkt auf der Timeline wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen alle Media-Typen für Element zulässig siehe Eigenschaft .canSeek
.seekSegmentTime()	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline ohne Wiederholung der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft .canSeek
.seekTo()	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft .isActive das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig



	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.seekToFrame()	Frame eines aktiven Elementes auf der Timeline anwählen
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.segmentTimeToActiveTime()	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
.segmentTimeToSimpleTime()	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
.setActive()	playItem Objekt als aktiven Track setzen und als solchen in der Behavior-Collection .style.time2.playlist registrieren
	verändert .isActive
	Track entspricht playItem Objekt (Playlisten-Eintrag)
	Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
	aktiver Track in der Liste: wird abgespielt
	Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten
.simpleTimeToSegmentTime()	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren

Events:

onbegin	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird
	nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Element
	erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft .repeatCount > 1 ist für das Eltern-Element
	siehe onend und onrepeat
onend	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut .repeatCount
	bzw. wenn das aktive Element gestoppt wird
	erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit das Kindelement ebenfalls endet
	nicht erzeugt wenn .fill mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht
	siehe Methode .endElement() und Eigenschaft .end
	siehe onbegin und onrepeat
onmediacomplete	erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde für Animation per Timeline
	Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc.
	sinnvoll für Start des Elementes auf der Timeline
	siehe onmediaerror
onmediaerror	erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält
	für Animation per Timeline
	Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc.
	sinnvoll für Start des Elementes auf der Timeline
	ersetzt das Ereignis onmedialoadfailed, das deprecated ist und nicht mehr verwendet werden darf !
	siehe onmediacomplete
onoutofsync	erzeugt wenn das Element seine Timeline verliert, also nicht mehr auf seiner Timeline animiert wird (nicht synchron zur Timeline des Elementes)
	sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked":
	EventHandler sollte neu synchronisieren z.B. Reset des Elementes durch .resetElement()
	siehe onsyncstored
onpause	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren
	auch bei body Objekt
onrepeat	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes
	nicht erzeugt beim Start des aktiven Elementes (siehe onbegin), also des ersten Durchlaufes, der keine Wiederholung ist
	nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für onrepeat kodiert hat (kein Heraufreichen des Ereignisses onrepeat zum Elternelement)
	.repeatCount bzw. .repeat muss > 1 sein:
	onrepeat wird also .repeatCount – 1 mal erzeugt
	Kind eines Elementes
	siehe onend, onbegin, .repeatCount und .repeat



onreset	erzeugt wenn Element zurückgesetzt wurde per Methode .resetElement(), also die Timeline des Elementes den aktuellen Wert laut .begin wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanzierung des Elementes
onresume	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
onreverse	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist) .autoReverse muss auf "true" stehen
onseek	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde: .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame()
onsyncrestored	erzeugt wenn Element gerade mit seiner Timeline synchronisiert wird nach vorausgegangenem Abbruch der Synchronisation siehe onoutofsync
ontrackchange	sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked" erzeugt wenn der Track einer Advanced Stream Redirector (ASX) Datei (*.asx) in der Playliste gewechselt wurde

4.3.2.2.4.3.39.23.9.3.14. *.style.time2.set Behavior-Objekt des Internet Explorer*

Timerobjekt zur Animation eines HTML-Elementes anhand eines Styles unter Verwendung der Attribute

ATTRIBUTENAME und **TARGETELEMENT**

Syntax:

XML	t:SET
Script	set

Beispiel:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:SET ID="ID_Set"
ATTRIBUTENAME="width"
BEGIN="2"
TO="300"
DUR="5"
TARGETELEMENT="ID_Div"
>
</t:SET>
<DIV ID="ID_Div"
STYLE="width:100px; height:50px;"
>
Test
</DIV>
</BODY>
</HTML>
```

Eigenschaften:

.attributeName	Bezeichner einer Style-Eigenschaft, also Name eines Attributes, für die Animation anhand dieses Attributes auf der Timeline
.autoReverse	siehe .accumulate .by .additive .calcMode automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.begin	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction
.dur	Dauer Objektaktivitäten laut Eigenschaft .timeAction alternativ: Eigenschaft .end
.end	Objektaktivitäten laut Eigenschaft .timeAction beenden ab IE 6.x alternativ: Eigenschaft .dur
.fill	Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber bevor die Timeline des Elternelementes endet ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet werden darf !!
.hasMedia	Objekt ist HTML-Media-Objekt
.repeatDur	Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline verlangt kodierte Eigenschaft .dur oder .repeat darf nicht kodiert werden mit der Eigenschaft .repeatCount
.restart	generelle Restartmöglichkeit eines Elementes auf der Timeline ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!



<code>.systemBitrate</code>	wird hier nicht erklärt
<code>.systemCaptions</code>	wird hier nicht erklärt
<code>.systemLanguage</code>	Sprache festlegen für das Objekt
<code>.systemOverdubOrSubtitle</code>	wird hier nicht erklärt
<code>.targetElement</code>	ID des zu animierenden Elementes auf der Timeline muss für Kindelement immer kodiert werden, wenn nicht das Elternelement animiert werden soll muss nicht kodiert werden, wenn kein Elternelement vorliegt Achtung: Objekt body ist das oberste Elternelement im Dokument und somit haben Elemente innerhalb BODY immer Eltern Empfehlung: immer kodieren ist der Bezug des Time-Container auf das zu animierende Element
<code>.timeContainer</code>	Typ der Timeline des Objektes
<code>.timeParent</code>	Zeiger auf das Eltern-Timeline
<code>.to</code>	Endwert zur Werterhöhung bei Elemente-Animation(en) auf der Timeline per <code>.additive</code> oder <code>.accumulate</code> für die Objekte <code>animate</code> , <code>animateMotion</code> und <code>animateColor</code> gilt: <code>.to</code> wird von <code>.path</code> und <code>.values</code> überschrieben <code>.to</code> überschreibt <code>.by</code> Achtung: Im Objekt <code>animatecolor</code> (<code>t:ANIMATECOLOR</code>) ist <code>.values</code> auch als Farbliste definiert !
Methoden:	
<code>.activeTimeToParentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.activeTimeToSegmentTime()</code>	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.beginElement()</code>	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
<code>.beginElementAt()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start
<code>.documentTimeToParentTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
<code>.endElement()</code>	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
<code>.endElementAt()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne
<code>.parentTimeToActiveTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren
<code>.parentTimeToDocumentTime()</code>	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
<code>.pauseElement()</code>	aktives Element auf Timeline pausieren lassen ersetzt Methode <code>pause()</code> , die deprecated ist und nicht mehr verwendet werden darf ! per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.resetElement()</code>	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
<code>.resumeElement()</code>	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben falls Element nicht pausiert, passiert nichts ersetzt die Methode <code>resume()</code> , da sie deprecated ist und nicht mehr verwendet werden darf per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.seekActiveTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Timeline wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen alle Media-Typen für Element zulässig siehe Eigenschaft <code>.canSeek</code>
<code>.seekSegmentTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline ohne Wiederholung der Animation



	wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler
<code>.seekTo()</code>	siehe Eigenschaft <code>.canSeek</code> aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler
<code>.seekToFrame()</code>	siehe Eigenschaft <code>.canSeek</code> Frame eines aktiven Elementes auf der Timeline anwählen wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler
<code>.segmentTimeToActiveTime()</code>	siehe Eigenschaft <code>.canSeek</code> Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
<code>.segmentTimeToSimpleTime()</code>	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
<code>.setActive()</code>	playItem Objekt als aktiven Track setzen und als solchen in der Behavior-Collection <code>.style.time2.playlist</code> registrieren verändert <code>.isActive</code> Track entspricht playItem Objekt (Playlisten-Eintrag) Trackliste liegt in der Behavior-Collection <code>.style.time2.playlist</code> als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
<code>.simpleTimeToSegmentTime()</code>	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei aktiver Track in der Liste: wird abgespielt Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren
Events:	
<code>onbegin</code>	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Element erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Eltern-Element siehe <code>onend</code> und <code>onrepeat</code>
<code>onend</code>	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut <code>.repeatCount</code> bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit das Kindelement ebenfalls endet nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code> siehe <code>onbegin</code> und <code>onrepeat</code>
<code>onpause</code>	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt
<code>onrepeat</code>	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe <code>onbegin</code>), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für <code>onrepeat</code> kodiert hat (kein Heraufreichen des Ereignisses <code>onrepeat</code> zum Elternelement) <code>.repeatCount</code> bzw. <code>.repeat</code> muss > 1 sein: <code>onrepeat</code> wird also <code>.repeatCount - 1</code> mal erzeugt Kind eines Elementes siehe <code>onend</code> , <code>onbegin</code> , <code>.repeatCount</code> und <code>.repeat</code>
<code>onreset</code>	erzeugt wenn Element zurückgesetzt wurde per Methode <code>.resetElement()</code> , also die Timeline des Elementes den aktuellen Wert laut <code>.begin</code> wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
<code>onresume</code>	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
<code>onreverse</code>	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also <code>.repeatCount > 0</code> ist) <code>.autoReverse</code> muss auf "true" stehen



4.3.2.2.4.3.39.23.9.3.15. .style.time2.seq Behavior-Objekt des Internet Explorer

Timerobjekt zur sequentiellen Animation

Syntax:

XML	t:SEQ
Script	seq

Beispiel 1:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
  <t:SEQ ID="ID_Seq"
    REPEATDUR="27"
    BEGIN="0"
  >
    <DIV ID="ID_Div1"
      CLASS="time_line_klasse"
      DUR="3"
    >
      3 Sekunden lang anzeigen
    </DIV>
    <DIV ID="ID_Div2"
      CLASS="time_line_klasse"
      DUR="4"
    >
      4 Sekunden lang anzeigen
    </DIV>
  </t:SEQ>
</BODY>
</HTML>

```

Beispiel 2:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
  <DIV STYLE="height:100px">
    <t:SEQ ID="ID_Seq" REPEATCOUNT="indefinite" >
      <t:MEDIA ID="ID_Media1"
        SRC = "test1.jpg"
        STYLE="position:absolute;"
        DUR="3"
        TIMECONTAINER="par"
        FILL="transition"
      >
        <t:TRANSITIONFILTER ID="ID_Transfilter1"
          TYPE="fade"
          DUR="2"
        >
        </t:TRANSITIONFILTER>
      </t:MEDIA>
      <t:MEDIA ID="ID_Media2"
        SRC = "test2.jpg"
        STYLE="position:absolute;"
        DUR="3"
        TIMECONTAINER="par"
        FILL="transition"
      >
        <t:TRANSITIONFILTER ID="ID_Transfilter2"
          TYPE="ClockWipe"
          DUR="2"
        >
        </t:TRANSITIONFILTER>
      </t:MEDIA>
    </t:SEQ>
  </DIV>
</BODY>
</HTML>

```



```

</DIV>
</BODY>
</HTML>

```

Beispiel 3:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:SEQ ID="ID_Seq"
CLASS="time_line_klasse"
>
<SPAN ID="ID_Span1"
CLASS="time_line_klasse"
TIMEACTION="display"
DUR="1"
>
5...
</SPAN>
<SPAN ID="ID_Span2"
CLASS="time_line_klasse"
TIMEACTION="display"
DUR="1"
>
4...
</SPAN>
<BR>
<IMG ID="ID_Img"
SRC="test.gif"
CLASS="time_line_klasse"
DUR="indefinite"
>
</t:SEQ>
<BR>
<BUTTON ID="ID_Button" onclick="ID_Seq.beginElement();">
Start der Timeline
</BUTTON>
</BODY>
</HTML>

```

Beispiel 4:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:SEQ ID="ID_Seq"
BEGIN="indefinite;"
>
<DIV ID="ID_Div1"
CLASS="time_line_klasse"
DUR="10"
style="position:relative;top:15px;left:25px;width:100px;height:100px;"
>
</DIV>
<DIV ID="ID_Div2"
CLASS="time_line_klasse"
DUR="10"
style="position:relative;top:30px;left:50px;width:100px;height:100px;"
>
</DIV>
<DIV ID="ID_Div3"
CLASS="time_line_klasse"
DUR="10"
style="position:relative;top:35px;left:75px;width:100px;height:100px;"
>

```



```

</DIV>
<DIV ID="ID_Div4"
      CLASS="time_line_klasse"
      DUR="10"
      style="position:relative;top:50px;left:100px;width:100px;height:100px;"
    >
</DIV>
</t:SEQ>
<BR>
<BUTTON ID="ID_Button1" onclick="ID_Seq.beginElement();"> start</BUTTON>
<BUTTON ID="ID_Button2" onclick="ID_Seq.nextElement();">next </BUTTON>
<BUTTON ID="ID_Button3" onclick="ID_Seq.prevElement();">previous</BUTTON>
<BUTTON ID="ID_Button4" onclick="ID_Seq.endElement();">stop</BUTTON>
</BODY>
</HTML>

```

Eigenschaften:

.accelerate	<p>Beschleunigung des Elementes auf der Timeline</p> <p>hat keinen Einfluss auf die Dauer der Timeline</p> <p>auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur</p> <p>Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten</p> <p>wenn ja, so werden beide Attribute ignoriert, also nicht verwendet</p> <p>siehe .decelerate</p>
.autoReverse	<p>automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline</p>
.begin	<p>Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction</p>
.decelerate	<p>Verlangsamung des Elementes auf der Timeline</p> <p>hat keinen Einfluss auf die Dauer der Timeline</p> <p>auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur</p> <p>Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten</p> <p>wenn ja, so werden beide Attribute ignoriert, also nicht verwendet</p> <p>siehe .accelerate</p>
.dur	<p>Dauer Objektaktivitäten laut Eigenschaft .timeAction</p>
.end	<p>alternativ: Eigenschaft .end</p> <p>Objektaktivitäten laut Eigenschaft .timeAction beenden</p> <p>ab IE 6.x</p> <p>alternativ: Eigenschaft .dur</p>
.fill	<p>Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber</p> <p>bevor die Timeline des Elternelementes endet</p> <p>ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet werden darf !!</p>
.hasMedia	<p>Objekt ist HTML-Media-Objekt</p>
.mediaDur	<p>Länge der Media-Datei in Sekunden (Länge bei der Wiedergabe) auf der Timeline</p>
.mediaHeight	<p>Ereignis onmediacomplete muss ausgelöst also die Media-Datei komplett geladen worden sein</p> <p>aktuelle Höhe des Media-Elementes in Pixels auf der Timeline</p> <p>Standardgemäß wird die Elementhöhe aus der Höhenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.height im Element kodieren)</p>
.mediaWidth	<p>aktuelle Breite des Media-Elementes in Pixels auf der Timeline</p> <p>Standardgemäß wird die Elementhöhe aus der Breitenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.width im Element kodieren)</p>
.mute	<p>Audio aktiv oder aus (stumm) auf der Timeline</p> <p>wenn Eigenschaft für body Objekt kodiert, so alle Audio-Elemente im Body davon betroffen</p> <p>beeinflusst nicht die Systemlautstärke-Regelung von Windows oder anderen Soundmixern</p> <p>siehe Objekt bgsound</p>
.repeatCount	<p>aktuelle Nummer der Wiederholung bei Loop</p>
.repeatDur	<p>Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline</p> <p>verlangt kodierte Eigenschaft .dur oder .repeat</p> <p>darf nicht kodiert werden mit der Eigenschaft .repeatCount</p>
.restart	<p>generelle Restartmöglichkeit eines Elementes auf der Timeline</p> <p>ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!</p>
.speed	<p>aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)</p>
.syncBehavior	<p>Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes, sinnvoll vorallem bei Time-Container (Gruppen von Objekten)</p> <p>Es muss die Eigenschaft .syncmaster kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der Beschreibung zum jeweiligen Objekt zu entnehmen.</p> <p>siehe .syncTolerance und .syncmaster</p>
.syncTolerance	<p>zeitliche Toleranz für Zwangs-Synchronisation von Elementen auf der Timeline</p> <p>also wenn Eigenschaft .syncBehavior auf "locked" gesetzt ist</p> <p>sinnvoll vorallem bei Time-Container (Gruppen von Objekten)</p> <p>siehe .syncBehavior und .syncMaster</p>
.systemBitrate	<p>wird hier nicht erklärt</p>
.systemCaptions	<p>wird hier nicht erklärt</p>
.systemLanguage	<p>Sprache festlegen für das Objekt</p>
.systemOverdubOrSubtitle	<p>wird hier nicht erklärt</p>
.timeAction	<p>Aktion des Objektes in der Timeline</p>



	Aktion nur ausführbar wenn Objekt aktiv ist UND Timeline aktiv ist
.timeContainer	Typ der Timeline des Objektes
.timeParent	Zeiger auf das Eltern-Timeline
.volume	Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes für alle Audio-Elemente des body Objektes auch möglich Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus ! siehe Objekt bgsound
Methoden:	
.activeTimeToParentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.activeTimeToSegmentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.beginElement()	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.beginElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start per Eigenschaft .isActive das Element auf Aktivsein prüfen
.documentTimeToParentTime()	Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
.endElement()	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.endElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne
.nextElement()	per Eigenschaft .isActive das Element auf Aktivsein prüfen nächstes Element aus aktiver t:SEQ animieren auf der Timeline wenn kein nächstes Element, so kein Fehler erzeugt
.nextTrack()	nächstes playItem Objekt aktivieren laut Behavior-Collection .style.time2.playlist also Abspielen des Tracks starten bzw. nächste Wiederholung starten wenn .repeatCount > 0 und noch nicht alle Wiederholungen abgearbeitet wurden wenn letzter Track laut der Playliste aktiv ist und dann der nächste Track aktiviert werden soll, so wird nicht der erste Track abgespielt, sondern der aktive Track in der Wiedergabe gestoppt und danach keine weiteren Aktionen mehr Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist. Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei aktiver Track in der Liste: wird abgespielt Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten
.parentTimeToActiveTime()	Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren
.parentTimeToDocumentTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
.pauseElement()	aktives Element auf Timeline pausieren lassen ersetzt Methode pause(), die deprecated ist und nicht mehr verwendet werden darf ! per Eigenschaft .isActive das Element auf Aktivsein prüfen
.prevElement()	vorhergehendes Element aus aktiver t:SEQ animieren auf der Timeline wenn kein vorhergehendes Element, so wird Timeline von t:SEQ auf 0 gesetzt
.prevTrack()	vorhergehendes playItem Objekt aktivieren laut Behavior-Collection .style.time2.playlist also Abspielen des Tracks starten aber nicht vorhergehende Wiederholung starten wenn .repeatCount > 0 und noch nicht alle Wiederholungen abgearbeitet wurden wenn erster Track laut der Playliste aktiv ist und dann der vorhergehende Track aktiviert werden soll, so wird der erste Track nochmal abgespielt



	Track entspricht playItem Objekt
	Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
	aktiver Track in der Liste: wird abgespielt
	Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten
.resetElement()	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
.resumeElement()	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben
	falls Element nicht pausiert, passiert nichts
	ersetzt die Methode resume(), da sie deprecated ist und nicht mehr verwendet werden darf
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
.seekActiveTime()	aktives Element animieren ab einem Zeitpunkt auf der Timeline
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	alle Media-Typen für Element zulässig
	siehe Eigenschaft .canSeek
.seekSegmentTime()	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline
	ohne Wiederholung der Animation
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.seekTo()	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline
	einschliesslich möglicher Wiederholungen der Animation
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.seekToFrame()	Frame eines aktiven Elementes auf der Timeline anwählen
	wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft .isActive das Element auf Aktivsein prüfen
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft .canSeek
.segmentTimeToActiveTime()	Wert der Segment-Timeline des Elements in den korrespondierenden Wert
	der aktiven Timeline des Elements konvertieren
.segmentTimeToSimpleTime()	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der
	Simple-Timeline des Elements konvertieren
.setActive()	playItem Objekt als aktiven Track setzen und als solchen in der Behavior-Collection .style.time2.playlist registrieren
	verändert .isActive
	Track entspricht playItem Objekt (Playlisten-Eintrag)
	Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste:
	Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
	aktiver Track in der Liste: wird abgespielt
	Track in der Liste auf aktiv setzen: Abspielen des Tracks sofort starten
.simpleTimeToSegmentTime()	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der
	Segment-Timeline des Elementes konvertieren
Events:	
onbegin	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird
	nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn
	Eigenschaft .repeatCount > 1 ist für das Element
	erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn
	Eigenschaft .repeatCount > 1 ist für das Eltern-Element
	siehe onend und onrepeat
onend	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive
	aller Wiederholungen laut .repeatCount
	bzw. wenn das aktive Element gestoppt wird
	erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat
	und somit des Kindelement ebenfalls endet
	nicht erzeugt wenn .fill mit Wert "freeze" oder "hold" für das Element kodiert wurde,
	es sei denn, das Elternelement hat das Ende seiner Timeline erreicht
	siehe Methode .endElement() und Eigenschaft .end
	siehe onbegin und onrepeat
onmediacomplete	erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde
	für Animation per Timeline
	Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer



	Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline siehe onmediaerror
onmediaerror	erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält für Animation per Timeline Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline ersetzt das Ereignis onmedialoadfailed, das deprecated ist und nicht mehr verwendet werden darf ! siehe onmediacomplete
onoutofsync	erzeugt wenn das Element seine Timeline verliert, also nicht mehr auf seiner Timeline animiert wird (nicht synchron zur Timeline des Elementes) sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked": Eventhandler sollte neu synchronisieren z.B. Reset des Elementes durch .resetElement() siehe onsyncrestored
onpause	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt
onrepeat	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe onbegin), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für onrepeat kodiert hat (kein Herausheben des Ereignisses onrepeat zum Elternelement) .repeatCount bzw. .repeat muss > 1 sein: onrepeat wird also .repeatCount - 1 mal erzeugt Kind eines Elementes siehe onend, onbegin, .repeatCount und .repeat
onreset	erzeugt wenn Element zurückgesetzt wurde per Methode .resetElement(), also die Timeline des Elementes den aktuellen Wert laut .begin wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
onresume	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
onreverse	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist) .autoReverse muss auf "true" stehen
onseek	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde: .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame()
onsyncrestored	erzeugt wenn Element gerade mit seiner Timeline synchronisiert wird nach vorausgegangenem Abbruch der Synchronisation siehe onoutofsync sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked"

4.3.2.2.4.3.39.23.9.3.16. *.style.time2.switch Behavior-Objekt des Internet Explorer*

Timerobjekt für automatische Auswahl und Aktivierung mindestens eines Elementes aus einer Menge von Elementen

Auswahl nur möglich anhand genau eines der folgenden systemnahen Attribute:

.systemBitrate
.systemCaptions
.systemLanguage
.systemOverdubOrSubtitle

das die Elemente unterstützen **und kodiert** haben **müssen**.

anhand der aktuellen zutreffenden Systemeinstellung:

Wenn der Attributwert mit der aktuellen Systemeinstellung übereinstimmt, so erfolgt die Auswahl und damit die
Abarbeitung **nur** des ausgewählten Elementes.

wird nicht angezeigt

nur von Elementen, die auf der gemeinsamen Hierarchie-Ebene **direkt unter** der von t:SWITCH kodiert wurden (Auswahl-
Ebene): Nur in der Auswahl-Ebene kodierte Attribute obiger Art werden für die Auswahl verwendet.

Element nur auswählbar, wenn auf der Hierarchie-Ebene **direkt unter** der von t:SWITCH kodiert wurde (Auswahl-Ebene):

Wenn Element ein HTML-Container ist, also HTML-Kinder hat, so werden in den Kindern eventuell kodierte Attribute
von obiger Art **nicht** in den Vergleich der Auswahl einbezogen, da die Kinder in einer tieferen Hierarchie-Ebene liegen
und nicht in der Auswahl-Ebene.

Aktivierung entspricht Abarbeitung des Elementes je nach Art des Elementes

Werden **nicht** obige Attribute unterstützt bzw. kodiert, so wird das Element **immer ausgewählt** (auch HTML-Kommentar),
also abgearbeitet !

wenn mit Attribut .timeContainer, also Element ist selbst ein Time-Container, so wird dieses Attribut ignoriert, wenn das Element
nicht automatisch ausgewählt wurde



Syntax:

XML t:SWITCH
Script switch

Beispiel für Auswahl anhand der aktuellen Sprache des Usersystems per Attribut **SYSTEMLANGUAGE**:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
  <t:SWITCH ID="ID_Switch">

    <t:SEQ ID="ID_Seq"
      STYLE="font-size:18pt;color:#ff0000"
    >
      <DIV ID="ID_Div1"
        CLASS="time_line_klasse"
        DUR="1.5"
      >
        Falls keine Sprachbezeichnung sichtbar,
      </DIV>
      <DIV ID="ID_Div2"
        CLASS="time_line_klasse"
        DUR="1.5"
      >
        so konnte nichts ausgewaehlt werden !
      </DIV>
    </t:SEQ>
    <BR>

    <SPAN ID="Span1"
      CLASS="time_line_Klasse"
      SYSTEMLANGUAGE="es"
    >
      Espanol
    </SPAN>

    <SPAN ID="Span2"
      CLASS="time_line_Klasse"
      SYSTEMLANGUAGE="en"
    >
      English
    </SPAN>

    <MARQUEE ID="ID_Marquee"
      SYSTEMLANGUAGE="de"
      LOOP=1
      HEIGHT=200
      WIDTH=200
      SCROLLAMOUNT=10
      SCROLLDELAY=20
      BEHAVIOR="SLIDE"
      DIRECTION="DOWN"
      STYLE="position:absolute; top:0; left:10"
    >
      Deutsch
    </MARQUEE>
    <BR>
    <SPAN ID="Span3"
      CLASS="time_line_Klasse"
      SYSTEMLANGUAGE="de"
    >
      wurde ausgewaehlt.
    </SPAN>

  </t:SWITCH>
</BODY>
</HTML>
```

Hinweise: Die Auswahl erfolgt anhand von Span1, Span2, MARQUEE und Span3 durch Vergleich auf die



aktuelle Systemsprache mit Spanisch (es) bzw. Englisch (en) bzw. Deutsch (de).
Ist das Usersystem deutschsprachig, so werden MARQUEE **und** Span3 aktiviert.
t:SEQ wird immer aktiviert, da kein SYSTEMLANGUAGE kodiert wurde.

Eigenschaften:

.hasMedia	Objekt ist HTML-Media-Objekt
.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt

Methoden:

keine

Events:

keine

4.3.2.2.4.3.39.23.9.3.17. .style.time2.transitionFilter Behavior-Objekt des Internet Explorer

Timerobjekt für **visuelle** Filteranimation als "Übergang"

zwischen 2 aktiven Elementen:

z.B. zwischen 2 Bilder, also Übergang vom Quell-Bild zum Ziel-Bild, wobei beide Bilder eine gemeinsame Position besitzen:

vor dem Start der Animation ist **mindestens** das Quell-Element sichtbar
nach dem Ende der Animation ist **mindestens** das Ziel-Element sichtbar

z.B. zwischen DIV und einem Bild, dass im DIV animiert wird: Bild taucht allmählich im DIV auf.

vor dem Start der Animation ist DIV ohne **oder** schon mit teilweiseem Bild
nach dem Ende der Animation ist DIV mit (auch teilweise) Bild

eines aktiven Elementes:

z.B. DIV, der sich wandelt (komplett oder teilweise)

Übergang:	von sichtbar zu unsichtbar:	vor Start des Filters sichtbar (auch teilweise) nach Start des Filters unsichtbar (auch teilweise)
oder	von unsichtbar zu sichtbar:	vor Start des Filters unsichtbar (auch teilweise) nach Start des Filters sichtbar (auch teilweise)

Element(e) müssen sichtbar sein bzw. sichtbar machbar sein, also aktiv sein.

Animation auch per Attribut TARGETELEMENT anhand des ID-Attributes des zu animierenden Elementes möglich.

Ein Element kann mit mehreren sequentiell kodierten transitionsFilter animiert werden, die z.B. innerhalb eines gemeinsamen Eltern-Time-Containers
oder von BODY
liegen.

Das Schreiben der transitionFilter-spezifischen Eigenschaften .by. .from .to und .value darf nur erfolgen, wenn der Übergangsfiler **nicht** aktiv ist. Also **erst** Filter deaktivieren, dann Eigenschaften belegen, **dann** Filter starten.

Bsp.:

```
<SCRIPT>
    object.endElement();
    object.by='0.4';
    object.beginElement();
</SCRIPT>
```

siehe auch Objekt filter

Syntax:

XML	t:TRANSITIONFILTER
Script	transitionFilter

Beispiel 1 Bild-Übergang innerhalb eines DIV:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
    <t:TRANSITIONFILTER ID="ID_Transfilter"
                        TYPE="fade"
                        DUR="8"
                        END="4"
                        TARGETELEMENT="ID_Div"
    >
    </ t:TRANSITIONFILTER>
    <DIV ID="ID_Div">
```



```

        CLASS="time_line_klasse"
        DUR="indefinite"
        STYLE="background-image:url(test.gif); background-repeat: no-repeat;"
    >
</DIV>
</BODY>
</HTML>

```

Beispiel 2 für einen DIV, der sich wandelt:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
    <DIV ID="ID_Div"
        CLASS="time_line_klasse"
        DUR="8"
        STYLE="width:270"
    >
        <t:TRANSITIONFILTER ID="ID_Transfilter1"
            BEGIN="ID_Div.begin"
            DUR="3"
            TYPE="clockWipe"
        >
        </t:TRANSITIONFILTER>

        <t:TRANSITIONFILTER ID="ID_Transfilter2"
            BEGIN="ID_Div.end - 3"
            DUR="3"
            MODE="out"
            TYPE="fade"
        >
        </t:TRANSITIONFILTER>

        Test
    </DIV>
</BODY>
</HTML>

```

Beispiel 3 für Animation eines Elementes mit mehreren transitionFilter, die in einem gemeinsamen Eltern-Container liegen

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
    <DIV ID="ID_Div" STYLE="height:100px">
        <t:SEQ ID="ID_Seq"
            REPEATCOUNT="indefinite"
        >
            <t:MEDIA ID="ID_Media1"
                SRC = "test1.jpg"
                STYLE="position:absolute; ....."
                DUR="3"
                TIMECONTAINER="par"
                FILL="transition"
            >
                <t:TRANSITIONFILTER ID="ID_Transfilter1"
                    TYPE="fade"
                    DUR="2"
                >
                </t:TRANSITIONFILTER>

                <t:TRANSITIONFILTER ID="ID_Transfilter2"
                    TYPE="ClockWipe"
                    DUR="2"
                >
                </t:TRANSITIONFILTER>
            </t:MEDIA>
        </t:SEQ>
    </DIV>

```



```

</t:TRANSITIONFILTER>

</t:MEDIA>
<t:MEDIA          ID="ID_Media2"
                  SRC = "test2.jpg"
                  STYLE="position:absolute; ...."
                  DUR="3"
                  TIMECONTAINER="par"
                  FILL="transition"
>
    <t:TRANSITIONFILTER  ID="ID_Transfilter3"
                        TYPE="ClockWipe"
                        DUR="2"
    >
    </t:TRANSITIONFILTER>
    <t:TRANSITIONFILTER  ID="ID_Transfilter4"
                        TYPE="fade"
                        DUR="2"
    >
    </t:TRANSITIONFILTER>

</t:MEDIA>
</t:SEQ>
</DIV>
</BODY>
</HTML>

```

Beispiel 4:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<DIV ID="ID_Div" STYLE="height:100px">
<t:SEQ  ID="ID_Seq"
      REPEATCOUNT="indefinite"
>
    <t:MEDIA          ID="ID_Media1"
                    SRC = "test1.jpg"
                    ALT="Test1"
                    STYLE="position:absolute; ...."
                    DUR="3"
                    TIMECONTAINER="par"
                    FILL="transition"
    >
        <t:TRANSITIONFILTER  ID="ID_Transfilter1"
                            TYPE="fade"
                            DUR="2"
        >
        </t:TRANSITIONFILTER>

        <t:TRANSITIONFILTER  ID="ID_Transfilter2"
                            TYPE="ClockWipe"
                            DUR="2"
        >
        </t:TRANSITIONFILTER>

    <MARQUEE          ID="ID_Marquee"
                    CLASS="time_line_klasse"
                    TIMECONTAINER="seq"
                    REPEATCOUNT="indefinite"
                    STYLE="position:absolute; ...."
    >
        <IMG          ID="ID_Img1"
                    SRC="test2.gif"
                    CLASS="time"
                    DUR="4"
                    ALT="Test2"
        >
        <IMG          ID="ID_Img2"
                    SRC="test3.gif"

```



```

        CLASS="time"
        DUR="4"
        ALT="Test3"
    >
    <IMG    ID="ID_Img3"
        SRC="test4.gif"
        CLASS="time"
        DUR="4"
        ALT="Test4"
    >
    <IMG    ID="ID_Img4"
        SRC="test5.gif"
        CLASS="time"
        DUR="4"
        ALT="Test5"
    >
    </MARQUEE>
</t:MEDIA>
</t:SEQ>
</DIV>
</BODY>
</HTML>

```

Beispiel 5 für 2 DIV, die sich unabhängig voneinander wandeln:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
    .time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
    <t:TRANSITIONFILTER    ID="ID_Transfilter1"
        TYPE="barWipe"
        DUR="3"
        TARGETELEMENT="ID_Div1"
        MODE="in"
    >
    </t:TRANSITIONFILTER>

    <t:TRANSITIONFILTER    ID="ID_Transfilter2"
        TYPE="barWipe"
        DUR="3"
        TARGETELEMENT="ID_Div2"
        MODE="out"
    >
    </t:TRANSITIONFILTER>

    <DIV    ID="ID_Div1"
        CLASS="time_line_klasse"
        BEGIN="0"
        DUR="indefinite"
        STYLE= "position:relative; left:20px; width:420px; height:100px;
            background-image:url(test.gif); background-repeat: no-repeat;
            "
    >
    </DIV>

    <DIV    ID="ID_Div2"
        CLASS="time_line_klasse"
        BEGIN="0"
        DUR="indefinite"
        STYLE= "position:relative; left:20px; width:420px; height:100px;
            background-image:url(test.gif); background-repeat: no-repeat;
            "
    >
    </DIV>
</BODY>
</HTML>

```

Beispiel 6 für mehrere transitionFilter innerhalb Body, die ein gemeinsames Element animieren:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">

```



```

<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>
<t:TRANSITIONFILTER ID="ID_Transfilter1"
TYPE="barWipe"
SUBTYPE="leftToRight"
DUR="3"
TARGETELEMENT="ID_Div1"
>
</t:TRANSITIONFILTER>

<t:TRANSITIONFILTER ID="ID_Transfilter2"
TYPE="starWipe"
SUBTYPE="fivePoint"
DUR="3"
TARGETELEMENT="ID_Div1"
>
</t:TRANSITIONFILTER>

<t:TRANSITIONFILTER ID="ID_Transfilter3"
TYPE="barnDoorWipe"
DUR="3"
TARGETELEMENT="ID_Div2"
FROM="0"
TO="1"
CALCMODE="linear"
MODE="in"
>
</t:TRANSITIONFILTER>

<DIV ID="ID_Div0" STYLE="height:170px;">

<DIV ID="ID_Div1"
CLASS="time_line_klasse"
STYLE="position:absolute; top:150px; left:20px; background-color:#3366CC;
padding:10px; height:80; color:white;
"
>
Test1
</DIV>

<DIV ID="ID_Div2"
CLASS="time_line_klasse"
STYLE="position:absolute; top:185px; left:60px; background-color:#FFCC00;
padding:10px; height:80;
"
>
Test2
</DIV>
</DIV>
</BODY>
</HTML>

```

Beispiel 7 für Start der Animation durch Ereignis onclick aus Klick auf ein Button:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<BODY>

<t:TRANSITIONFILTER ID="ID_Transfilter1"
BEGIN="ID_Div1.begin"
TYPE="barWipe"
DUR="5"
TARGETELEMENT="ID_Div1"
VALUES=".1;.17;.27;.37;.47;.56;.65;.71;.82;.92;1.0"

```



```

        CALCMODE="discrete"
    >
</t:TRANSITIONFILTER>

<t:TRANSITIONFILTER ID="ID_Transfilter2"
    BEGIN="ID_Div1.begin"
    TYPE="barWipe"
    DUR="5"
    TARGETELEMENT="ID_Div2"
    VALUES=".1;.17;.27;.37;.47;.56;.65;.71;.82;.92;1.0"
    CALCMODE="linear"
>
</t:TRANSITIONFILTER>
<BR>
<INPUT TYPE="button" ID="ID_Button" VALUE="Start Transition">

<DIV ID="ID_Div1"
    CLASS="time_line_Klasse"
    BEGIN="ID_Button.click"
    DUR="indefinite"
    STYLE="position:relative; left:20px; width:420px; height:100px;
        background-image:url(test.gif); background-repeat: no-repeat;
    "
>
</DIV>

<DIV ID="ID_Div2"
    CLASS="time_line_Klasse"
    BEGIN="ID_Button.click"
    DUR="indefinite"
    STYLE="position:relative; left:20px; width:420px; height:100px;
        background-image:url(test.gif); background-repeat: no-repeat;
    "
>
</DIV>
</BODY>
</HTML>

```

Eigenschaften:

Das Schreiben der transitionFilter-spezifischen Eigenschaften .by, .from, .to und .value darf nur erfolgen, wenn der Übergangsfiler **nicht** aktiv ist. Also **erst** Filter deaktivieren, dann Eigenschaften belegen, **dann** Filter starten.

Bsp.:

```

<SCRIPT>
    object.endElement();
    object.by=0.4;
    object.beginElement();
</SCRIPT>

```

.accelerate

Beschleunigung des Elementes auf der Timeline
 hat keinen Einfluss auf die Dauer der Timeline
 auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur
 Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten
 wenn ja, so werden beide Attribute ignoriert, also nicht verwendet
 siehe .decelerate

.accumulate

Kumulative Animation eines Elementes auf der Timeline
 Element darf nicht aktiv sein, wenn .accumulate definiert werden soll:
 Element erst danach starten.
 Es kann jede numerische Style-Eigenschaft kumulativ verändert werden und damit
 die kumulative Animation des Elementes in der Style-Eigenschaft erzeugen.
 Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des
 Wertes sein (mathematische Berechnungen erst nach Entfernung der Einheit
 per Stringoperationen möglich)
 Name der Eigenschaft laut .attributeName
 Änderung des Wertes pro Durchlauf bzw. kumulativ um den Maximalwert laut .by
 Anzahl der Durchläufe bzw. der Wert-Kumulationen laut .repeatCount
 Kumulationsschrittweite laut .calcMode
 Ereignis des Startes eines Durchlaufes bzw. Kumulation um Maximalwert ist onrepeat.
 Wert der Eigenschaft .repeatCount muss > 1 sein
 Kumulation nicht aktiv:
 Nach jedem Durchlauf laut .repeatCount erfolgt Rücksetzen der Style-Eigenschaft des
 Elementes auf den Zustand vor dem Durchlaufbeginn.
 Es wird also die Style-Eigenschaft nicht wertmäßig kumuliert.
 Das Element wird also laut .repeatCount mehrmals animiert.
 Pro Durchlauf wird für die Eigenschaft laut .attributeName der Wert



	ab Startwert in Schrittweite laut .calcMode um den maximalen Wert laut .by
	erhöht.
	Kumulation aktiv: Die Style-Eigenschaft des Elementes wird pro Durchlauf wertmäßig laut .repeatCount um den Wert laut .by kumuliert. Kumulationsschrittweite laut .calcMode Das Element wird also genau 1 mal animiert. Mit Beginn des Startes des Elementes wird für die Eigenschaft laut .attributName der Wert ab Startwert in Schrittweite laut .calcMode um den maximalen Wert aus dem Produkt aus .by mal Anzahl der Wiederholungen z.B. laut .repeatCount erhöht. Es wird also über alle Wiederholungen der Wert kumuliert.
.additive	siehe .additive .calcMode numerische Werte von Eigenschaften eines Elementes ersetzen oder zu den numerischen Werten der gleichnamigen Eigenschaften anderer Elemente addieren während der Elemente-Animation auf der Timeline Wert der Eigenschaft: numerisch aber Einheiten wie px, in, cm, mm, pt, pc können Teil des Wertes sein gültige Eigenschaften sind z.B. .values oder .by sinnvoll für Kombinationen von Elementen (auch bei Wiederholung eines Elementes) Element darf nicht aktiv sein, wenn .additive definiert werden soll: Element erst danach starten. Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert ! siehe .accumulate .by .attributeName
.autoReverse	automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.begin .by	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction Schrittweite des Übergangsfilters per .style.time2.transitionFilter Behavior-Objekt als Offset zum Wert der kodierten Eigenschaft .from des Behavior Schrittweite als Prozentsatz des Überganges nach dessen Vollendung nicht zusammen mit Eigenschaft .to oder .value kodieren, sonst wird .by ignoriert
.calcMode	Interpolation der Aufteilung der internen Animationsschritte auf Schrittweite laut .by Schritt laut .from bei optionalen .to Gesamtschritt bei fehlendem .by bzw. .from .to Animationsschritte laut .values per Übergangfilter per .style.time2.transitionFilter Behavior-Objekt Bsp.: Lineare Aufteilung durch Interpolation: Interne Schritte zu gleichen Teilen verteilt keine Interpolation, so Animation laut Schrittweite, also eventuell nicht fließende Animation Größe des internen Animationsschrittes: Je nach der Gesamtdauer der Animation laut .dur des Behavior Verlangsamung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .accelerate
.dur .end	zeitliche Gesamtdauer des Übergangsfilters per .style.time2.transitionFilter Behavior-Objekt Objektaktivitäten laut Eigenschaft .timeAction beenden ab IE 6.x alternativ: Eigenschaft .dur
.from	Start-Prozentsatz des kompletten Überganges bei Start der Animation des Übergangsfilters per .style.time2.transitionFilter Behavior-Objekt Start des Übergangsfilters mit bereits teilweise vollendetem Übergang nicht zusammen mit Eigenschaft .value kodieren, sonst wird .from ignoriert
.keySpline	Wert eines Pixel-Intervalles zur Werterhöhung bei 2D-Elemente-Animation(en) auf der Timeline per .additive oder .accumulate benötigt .calcMode auf "spline" .keyTimes .values oder .path
.keyTimes	Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert ! Zeitwert eines Pixel-Intervalles zur Werterhöhung bei 2D-Elemente-Animation(en) auf der Timeline per .additive oder .accumulate benötigt .calcMode auf "spline" .keySpline .values oder .path
.mode	Achtung: Im Objekt animatecolor (t:ANIMATECOLOR) ist .values auch als Farbliste definiert ! Ergebnis der Animation des Übergangsfilters per .style.time2.transitionFilter Behavior-Objekt, also Sichtbarkeit bzw. Unsichtbarkeit nach Vollendung des Filters
.repeatCount .repeatDur	aktuelle Nummer der Wiederholung bei Loop Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline



	verlangt kodierte Eigenschaft .dur oder .repeat darf nicht kodiert werden mit der Eigenschaft .repeatCount
.restart	generelle Restartmöglichkeit eines Elementes auf der Timeline ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!
.speed	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
.subtype	visuelle Erscheinungsform des Überganges per .style.time2.transitionFilter Behavior-Objekt, also visuelle Art und Weise des Überganges
.targetElement	optional kodierbar zur Eigenschaft .type des Behavior ID des zu animierenden Elementes auf der Timeline muss für Kindelement immer kodiert werden, wenn nicht das Elternelement animiert werden soll muss nicht kodiert werden, wenn kein Elternelement vorliegt Achtung: Objekt body ist das oberste Elternelement im Dokument und somit haben Elemente innerhalb BODY immer Eltern Empfehlung: immer kodieren
.to	ist der Bezug des Time-Container auf das zu animierende Element Ende-Prozentsatz des kompletten Überganges bei Ende der Animation des Übergangsfilters per .style.time2.transitionFilter Behavior-Objekt Ende des Übergangsfilters mit nur teilweise vollendetem Übergang nicht zusammen mit Eigenschaften .by und .value kodieren, sonst wird .to ignoriert
.type	visuelle Erscheinungsform des Überganges per .style.time2.transitionFilter Behavior-Objekt, also visuelle Art und Weise des Überganges
.values	optional kodierbar ist zusätzlich die Eigenschaft .subtype des Behavior Animationsschritte als Prozentsatz des kompletten Überganges per .style.time2.transitionFilter Behavior-Objekt nicht zusammen mit Eigenschaften .by und .to und .from kodieren, da diese sonst ignoriert werden

Methoden:

.activeTimeToParentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.activeTimeToSegmentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.beginElement()	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes..
.beginElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start
.documentTimeToParentTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
.endElement()	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes.
.endElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne
.parentTimeToActiveTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des Elementes konvertieren
.parentTimeToDocumentTime()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
.pauseElement()	aktives Element auf Timeline pausieren lassen ersetzt Methode pause(), die deprecated ist und nicht mehr verwendet werden darf ! per Eigenschaft .isActive das Element auf Aktivsein prüfen
.resumeElement()	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben falls Element nicht pausiert, passiert nichts ersetzt die Methode resume(), da sie deprecated ist und nicht mehr verwendet werden darf per Eigenschaft .isActive das Element auf Aktivsein prüfen
.seekActiveTime()	aktives Element animieren ab einem Zeitpunkt auf der Timeline wenn Element nicht aktiv, so Fehler erzeugt



	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen alle Media-Typen für Element zulässig siehe Eigenschaft <code>.canSeek</code>
<code>.seekSegmentTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline ohne Wiederholung der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.seekTo()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen nicht alle Media-Typen für Element zulässig wenn unzulässig, so kein Fehler siehe Eigenschaft <code>.canSeek</code>
<code>.segmentTimeToActiveTime()</code>	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
<code>.segmentTimeToSimpleTime()</code>	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
<code>.simpleTimeToSegmentTime()</code>	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren
Events:	
<code>onbegin</code>	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Element erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Eltern-Element siehe <code>onend</code> und <code>onrepeat</code>
<code>onend</code>	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut <code>.repeatCount</code> bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit das Kindelement ebenfalls endet nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code> siehe <code>onbegin</code> und <code>onrepeat</code>
<code>onpause</code>	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt
<code>onrepeat</code>	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe <code>onbegin</code>), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für <code>onrepeat</code> kodiert hat (kein Heraufreichen des Ereignisses <code>onrepeat</code> zum Elternelement) <code>.repeatCount</code> bzw. <code>.repeat</code> muss > 1 sein: <code>onrepeat</code> wird also <code>.repeatCount - 1</code> mal erzeugt Kind eines Elementes siehe <code>onend</code> , <code>onbegin</code> , <code>.repeatCount</code> und <code>.repeat</code>
<code>onresume</code>	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
<code>onreverse</code>	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also <code>.repeatCount > 0</code> ist) <code>.autoReverse</code> muss auf "true" stehen
<code>onseek</code>	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde: <code>.seekActiveTime()</code> <code>.seekSegmentTime()</code> <code>.seekTo()</code> <code>.seekToFrame()</code>

4.3.2.2.4.3.39.23.9.3.18. *.style.time2.video Behavior-Objekt des Internet Explorer*

Timerobjekt zur Wiedergabe von Video

Syntax:

```
XML      t:VIDEO
Script   video
```

Beispiel 1:

```
<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
```



```

.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
</HEAD>
<SCRIPT>
    function Aendern()
    {
        ID_Video.updateMode = ID_Select1.options.value;
        ID_Video.speed      = ID_Select1.options.value;
    }
</SCRIPT>
</HEAD>
<BODY>
    <t:VIDEO          ID="ID_Video"
                     SRC="test.avi"
                     UPDATEMODE="reset"
                     STYLE="width:175px; height:150px;"
    >
    </t:VIDEO>
    <BR>
    updateMode waehlen:
    <SELECT NAME="ID_Select1">
        <OPTION VALUE="auto">Auto</OPTION>
        <OPTION VALUE="reset" SELECTED>Reset</OPTION>
    </SELECT>
    <BR>
    Geschwindigkeit waehlen:
    <SELECT NAME="ID_Select2">
        <OPTION VALUE="0.25">25%</OPTION>
        <OPTION VALUE="0.50">50%</OPTION>
        <OPTION VALUE="0.75">75%</OPTION>
        <OPTION VALUE="1" SELECTED>100% </OPTION>
        <OPTION VALUE="2" SELECTED>200% </OPTION>
    </SELECT>
    <BR>
    <BUTTON ID="ID_Button1" onClick="Aendern();">
        Geschwindigkeit aendern
    </BUTTON>
    <BUTTON ID="ID_Button1" onClick="document.body.beginElement()">
        Restart
    </BUTTON>
</CENTER>
</BODY>
</HTML>

```

Beispiel 2:

```

<HTML XMLNS:t="urn:schemas-microsoft-com:time">
<HEAD>
<?IMPORT namespace="t" implementation="#default#time2">
<STYLE>
.time_line_klasse { behavior: url(#default#time2) }
</STYLE>
<SCRIPT>
    function Anzeige()
    {
        switch (ID_Video.currTimeState.state)
        {
            case 1:
                if (ID_Video.currTimeState.isPaused == true)
                {
                    ID_Button2.disabled = true;
                    ID_Button3.disabled = true;
                    ID_Button4.disabled = false;
                    ID_Button5.disabled = false;
                }
                else
                {
                    ID_Button2.disabled = true;
                    ID_Button3.disabled = false;
                    ID_Button4.disabled = true;
                    ID_Button5.disabled = false;
                }
                break;
            case 0:
                ID_Button2.disabled = false;

```



```

        ID_Button3.disabled = true;
        ID_Button4.disabled = true;
        ID_Button5.disabled = true;
        break;
    case 4:
        ID_Button2.disabled = false;
        ID_Button3.disabled = true;
        ID_Button4.disabled = true;
        ID_Button5.disabled = true;
        break;
    }
}
</SCRIPT>
<SCRIPT FOR="document" EVENT="onclick" LANGUAGE="JScript">
    Anzeige();
</SCRIPT>
</HEAD>
<BODY onload=" Anzeige()">
    <t:VIDEO ID="ID_Video"
        CLASS="time_line_klasse"
        BEGIN="indefinite"
        STYLE="position:absolute;top:90px;height:150px;"
        FILL="remove"
        SRC="/test /media/movie.avi"
    >
    </ t:VIDEO>
    <SPAN ID="ID_Span"
        STYLE="position:absolute;top:255px;"
    >
        Status: 0
    </SPAN>
    <P STYLE="position:absolute;top:280px;">
        <BUTTON ID="ID_Button1"
            onclick=
                "ID_Span.innerText='Status: ' + ID_Video.currTimeState.state"
        >
            Aktueller Status
        </BUTTON>
        <BUTTON ID="ID_Button2"
            onclick="ID_Video.beginElement();"
        >
            Beginn
        </BUTTON>
        <BUTTON ID="ID_Button3"
            onclick="ID_Video.pauseElement();"
        >
            Pause
        </BUTTON>
        <BUTTON ID="ID_Button4"
            onclick="ID_Video.resumeElement();"
        >
            Resume
        </BUTTON>
        <BUTTON ID="ID_Button5"
            onclick=
                "ID_Video.endElement();ID_Span.innerText='Status: 0'"
        >
            Stop
        </BUTTON>
    </P>
</BODY>
</HTML>

```

Eigenschaften:

.abstract	Beschreibung einer Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei wird der Text unter ABSTRACT vom aktuellen Eintrag geliefert und nicht der Datei selbst
.accelerate	Beschleunigung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .decelerate
.author	Name des Autor der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von AUTHOR des aktiven



	Eintrages geliefert und nicht den der Datei.
	Track entspricht playItem Objekt
	Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.autoReverse	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei automatische Rückwärtsanimation eines Elementes nach dem kompletten Ende der Vorwärtsanimation des Elementes auf der Timeline
.Banner	Text laut BANNER des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.BannerAbstract	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei Text laut BANNER und dort ABSTRACT des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.BannerMoreInfo	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei Text laut BANNER und dort MOREINFO des aktiven Eintrages in einer Advanced Stream Redirector (ASX) –Datei anhand Behavior-Collection .style.time2.playList ab IE 6.x Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.begin	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
BOUNDARY	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction
.bufferingProgress	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software aktueller prozentualer Status des Pufferns eines Datenflusses (Data Stream) einer Media-Datei auf der Timeline Prozentwert des bisher erfolgten Pufferns nur für Media-Datei mit Datenfluss
.canPause	generelle Pausierungsmöglichkeit einer Media-Datei auf der Timeline
.canSeek	generelle Möglichkeit der Auswahl eines Zeitpunktes einer Media-Datei auf der Timeline Auswahl per Seek-Methoden .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame()
.clipBegin	nicht jeder Media-Typ unterstützt Seek Startzeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline Eigenschaft .canSeek muss true liefern
.clipEnd	nicht jeder Media-Typ unterstützt Clipping Endzeitpunkt eines Unterbereiches (Clip) innerhalb des Media-Elementes auf der Timeline Eigenschaft .canSeek muss true liefern
.copyright	nicht jeder Media-Typ unterstützt Clipping Copyright der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von COPYRIGHT des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playList als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
.currentFrame	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei Nummer des aktuellen Frame einer Media-Datei auf der Timeline
.decelerate	nicht alle Media-Typen unterstützen Frames Verlangsamung des Elementes auf der Timeline hat keinen Einfluss auf die Dauer der Timeline auch wirksam bei Wiederholungen per Attribute .repeatCount oder .repeatDur Summe der Werte der Attribute .accelerate und .decelerate darf nicht 1 überschreiten wenn ja, so werden beide Attribute ignoriert, also nicht verwendet siehe .accelerate
.downloadCurrent	siehe .accelerate Anzahl der bisher downgeloadeten Bytes beim Laden einer Media-Datei auf der Timeline (Laden des Daten-Stream in Form der Media-Datei) nur für Media-Datei mit Datenfluss
.downloadTotal	Anzahl der insgesamt downgeloadeten Bytes nach dem kompletten Laden einer Media-Datei auf der Timeline (Laden des Daten-Stream in Form der Media-Datei)



.dur	Dauer Objektaktivitäten laut Eigenschaft .timeAction alternativ: Eigenschaft .end
.end	Objektaktivitäten laut Eigenschaft .timeAction beenden ab IE 6.x alternativ: Eigenschaft .dur
.fill	Aktion eines Elementes zum Zeitpunkt des Ende der Timeline des Elementes aber bevor die Timeline des Elternelementes endet ist Ersatz für die Eigenschaft endHold, die deprecated ist und nicht mehr verwendet werden darf !!
.hasAudio	Media-Datei mit Audioinhalt auf der Timeline nicht alle Media-Typen können Audio beinhalten Stummschaltung oder Lautstärkeeinstellungen sind dabei egal
.hasDownloadProgres	Start des Donloades einer Media-Datei auf der Timeline
.hasMedia	Objekt ist HTML-Media-Objekt
.hasPlayList	Verfügbarkeit der Behavior-Collection .style.time2.playList für Element auf der Timeline, also ob Element eine Playliste hat
.hasVisual	Media-Datei mit visuellem Inhalt auf der Timeline visuelle Daten werden auf dem Bildschirm sichtbar nicht alle Media-Typen können sichtbare Daten beinhalten
IMMEDIATEEND	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.isStreamed	Media-Datei mit Datenfluss (Data Stream) auf der Timeline nicht alle Media-Typen unterstützen Datenstrom
.latestMediaTime	Wartezeit für Start der Wiedergabe eines Media-Elementes auf der Timeline nicht für Media-Datei mit Datenfluss
LONGTRANSITION	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mediaDur	Länge der Media-Datei in Sekunden (Länge bei der Wiedergabe) auf der Timeline Ereignis onmediacomplete muss ausgelöst also die Media-Datei komplett geladen worden sein
.mediaHeight	aktuelle Höhe des Media-Elementes in Pixels auf der Timeline Standardgemäß wird die Elementhöhe aus der Höhenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.height im Element kodieren)
.mediaWidth	aktuelle Breite des Media-Elementes in Pixels auf der Timeline Standardgemäß wird die Elementhöhe aus der Breitenangabe in der Media-Datei verwendet (ansonsten STYLE-Angabe zu .style.width im Element kodieren)
.mimeType	MIME-Typ (Multipurpose Internet Mail Extension-Typ) des Media-Elementes auf der Timeline
MODULATE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
MOTIFNAME	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.mute	Audio aktiv oder aus (stumm) auf der Timeline wenn Eigenschaft für body Objekt kodiert, so alle Audio-Elemente im Body davon betroffen beeinflusst nicht die Systemlautstärke-Regelung von Windows oder anderen Soundmixern siehe Objekt bgsound
.player	Wiedergabeplayer einem Media-Element zuordnen für Wiedergabe auf der Timeline Player ist eine Softwarekomponente, die im System und im Browser installiert sein muss ab IE 6.x werden keine Plugins mehr unterstützt sondern nur noch ActiveX-Controls z.B. ist {22d6f312-b0f6-11d0-94ab-0080c74c7e95} das ActiveX-Control für den Windows Media-Player im Internet Explorer.
.playerObject	Zeiger auf den Wiedergabeplayer, der das Media-Element auf der Timeline wiedergibt besonders zu verwenden, wenn Player über ActiveX-Control im Browser implementiert ist Zeiger ermöglicht, die Methoden und Eigenschaften des Players anzusprechen siehe Playerbeschreibung des Herstellers vom Player (im Falle von Microsoft sind die Methoden und Eigenschaften im jeweiligen SDK der Player-Version zu finden)
.rating	Playerart laut .player
.repeatCount	Rating der Media-Datei auf der Timeline
.repeatDur	aktuelle Nummer der Wiederholung bei Loop Anzahl der Dauer in Sekunden für Wiederholung auf der Timeline verlangt kodierte Eigenschaft .dur oder .repeat darf nicht kodiert werden mit der Eigenschaft .repeatCount
.restart	generelle Restartmöglichkeit eines Elementes auf der Timeline ersetzt die Eigenschaft eventRestart, da diese deprecated ist und nicht mehr verwendet werden darf !!
SEGMENTTYPE	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software
.speed	aktuelle Wiedergabegeschwindigkeit des Kindes (Run time speed)
.src	Url einer Media-Datei auf der Timeline
.syncBehavior	Synchronisation der Timeline eines Elementes mit der Timeline des Elternelementes, sinnvoll vorallem bei Time-Container (Gruppen von Objekten) Es muss die Eigenschaft .syncmaster kodiert sein. Ob das Element diese Eigenschaft unterstützt, ist der Beschreibung zum jeweiligen Objekt zu entnehmen. siehe .syncTolerance und .syncmaster
.syncMaster	Synchronisierung der Animation des im Container liegenden Elementes auf Timeline (Container ist Master in der Synchronisierung), wobei ein Master nur genau ein zu synchronisierendes Element haben darf (Eineindeutigkeit). nur sinnvoll bei Zwangssynchronisierung (siehe .syncBehavior) ersetzt die Eigenschaft clockSource, da diese deprecated ist und nicht mehr verwendet werden darf ! siehe .syncTolerance und .syncBehavior
.syncTolerance	zeitliche Toleranz für Zwangs-Synchronisation von Elementen auf der Timeline



	also wenn Eigenschaft .syncBehavior auf "locked" gesetzt ist sinnvoll vor allem bei Time-Container (Gruppen von Objekten) siehe .syncBehavior und .syncMaster
.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.timeAction	Aktion des Objektes in der Timeline Aktion nur ausführbar wenn Objekt aktiv ist UND Timeline aktiv ist
.timeContainer	Typ der Timeline des Objektes
.timeParent	Zeiger auf das Eltern-Timeline
.title	Titel der Media-Datei auf der Timeline bei Advanced Stream Redirector (ASX)-Datei: Es wird der Text von TITLE des aktiven Eintrages geliefert und nicht den der Datei. Track entspricht playItem Objekt Trackliste liegt in der Behavior-Collection .style.time2.playlist als Playliste: Diese ist nur referenzierbar, solange das Elternobjekt des Elementes auf der Eltern-Timeline aktiv ist, also das Elternobjekt noch nicht auf seiner Timeline geendet ist.
TRANSITIONTYPE	Advanced Stream Redirector (ASX) –Datei: Playlisten-Datei
.type	wird hier nicht erklärt, da Komponenten von MS DirectMusic-Software MIME-Typ eines Media-Elementes auf der Timeline Media-Datei zum Element laut Eigenschaft .src
.updateMode	Art des Updates der Eigenschaften eines Elementes nach dem Start des Elementes auf der Timeline folgende Eigenschaften können geupdatet werden: .autoReverse .begin .dur .end .fill .repeatCount .repeatDur .speed
.URL	Url aus einem Scriptkommando aus einer Advanced Streaming Format (ASF)-Datei es muss Event onURLFlip erzeugt worden sein
.volume	Lautstärke eines Audio-Elementes auf der Timeline relativ zur Lautstärke des Elternelementes für alle Audio-Elemente des body Objektes auch möglich Achtung: Bei Verwendung von DirectMusic wirkt sich das auf den DirectMusic-Player aus ! siehe Objekt bgsound

Methoden:

.activeTimeToParentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline der Eltern konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.activeTimeToSegmentTime()	Wert der aktiven Timeline des Elementes in den korrespondierenden Wert der Timeline des Segmentes konvertieren per Eigenschaft .isActive das Element auf Aktivsein prüfen
.beginElement()	Element auf der Timeline starten, also aktivieren identisch mit Wirkung, wenn Startzeit des Elementes erreicht wird auf der Timeline Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes.
.beginElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen Element auf der Timeline zu einem spezifischen Zeitpunkt ab Start der Timeline starten (aktivieren) also mit Wartezeit ab Beginn der Timeline des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so startet das Kindelement sofort, also ohne Wartezeit wenn zusätzlich ein weiterer Start (oder mehrere Starts) mit identischem oder kleinerem Zeitpunkt kodiert, so wird dieser zusätzliche ignoriert mit größerem Zeitpunkt kodiert, so wird dieser Start auch ausgeführt also Neustart nach einem erfolgten Start per Eigenschaft .isActive das Element auf Aktivsein prüfen
.documentTimeToParentTime()	Wert der Timeline des Dokumentes in den korrespondierenden Wert der Timeline der Eltern des Elementes konvertieren
.endElement()	aktives Element auf der Timeline stoppen identisch mit Wirkung, wenn Endezeit des Elementes erreicht wird auf der Timeline bzw. die Dauer (Duration) des Elementes abgearbeitet wurde Alle Kinder des Elementes erhalten Information über Start des Elternelementes und sind somit korrekt auf der Timeline des Elternelementes.
.endElementAt()	per Eigenschaft .isActive das Element auf Aktivsein prüfen aktives Element auf der Timeline nach einer Zeitspanne der Aktivität des Elementes ab Beginn der Timeline stoppen sinnvoll z.B. vor Erreichung des Ende der Dauer (Duration) des Elementes wenn das Elternelement diesen Zeitpunkt schon erreicht hat, so endet das Kindelement sofort, also ohne Zeitspanne per Eigenschaft .isActive das Element auf Aktivsein prüfen
.parentTimeToActiveTime()	Wert der Timeline der Eltern zum korrespondierenden Wert der aktiven Timeline des



	Elementes konvertieren
	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.parentTimeToDocumentTime()</code>	Wert der Timeline der Eltern des Elementes in den korrespondierenden Wert der Timeline des Dokumentes konvertieren
<code>.pauseElement()</code>	aktives Element auf Timeline pausieren lassen ersetzt Methode <code>pause()</code> , die deprecated ist und nicht mehr verwendet werden darf !
	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.resetElement()</code>	alle Timeline-Einstellungen zum Element aufheben (egal ob Element aktiv oder nicht)
<code>.resumeElement()</code>	Pause eines aktiven Elementes, das auf der Timeline pausiert, aufheben falls Element nicht pausiert, passiert nichts ersetzt die Methode <code>resume()</code> , da sie deprecated ist und nicht mehr verwendet werden darf
	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
<code>.seekActiveTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Timeline wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
	alle Media-Typen für Element zulässig
	siehe Eigenschaft <code>.canSeek</code>
<code>.seekSegmentTime()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline ohne Wiederholung der Animation wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
	Software-Player zum Mediatyp des Elementes wird nicht davon beeinflusst
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft <code>.canSeek</code>
<code>.seekTo()</code>	aktives Element animieren ab einem Zeitpunkt auf der Segment-Timeline einschliesslich möglicher Wiederholungen der Animation wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft <code>.canSeek</code>
<code>.seekToFrame()</code>	Frame eines aktiven Elementes auf der Timeline anwählen wenn Element nicht aktiv, so Fehler erzeugt
	per Eigenschaft <code>.isActive</code> das Element auf Aktivsein prüfen
	nicht alle Media-Typen für Element zulässig
	wenn unzulässig, so kein Fehler
	siehe Eigenschaft <code>.canSeek</code>
<code>.segmentTimeToActiveTime()</code>	Wert der Segment-Timeline des Elements in den korrespondierenden Wert der aktiven Timeline des Elements konvertieren
<code>.segmentTimeToSimpleTime()</code>	Wert der Segment-Timeline eines Elements in den korrespondierenden Wert der Simple-Timeline des Elements konvertieren
<code>.simpleTimeToSegmentTime()</code>	Wert der Simple-Timeline eines Elementes in den korrespondierenden Wert der Segment-Timeline des Elementes konvertieren

Events:

<code>onbegin</code>	erzeugt wenn ein Element auf der Timeline aktiviert und damit die Timeline aktiv wird nicht erzeugt für eine Wiederholung der Animation des Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Element erzeugt für eine Wiederholung der Animation des Eltern-Elementes, also wenn Eigenschaft <code>.repeatCount > 1</code> ist für das Eltern-Element siehe <code>onend</code> und <code>onrepeat</code>
<code>onend</code>	erzeugt wenn das aktive Element das Ende der aktiven Timeline erreicht inklusive aller Wiederholungen laut <code>.repeatCount</code> bzw. wenn das aktive Element gestoppt wird erzeugt wenn das Elternelement das Ende seiner Timeline erreicht hat und somit das Kindelement ebenfalls endet nicht erzeugt wenn <code>.fill</code> mit Wert "freeze" oder "hold" für das Element kodiert wurde, es sei denn, das Elternelement hat das Ende seiner Timeline erreicht siehe Methode <code>.endElement()</code> und Eigenschaft <code>.end</code> siehe <code>onbegin</code> und <code>onrepeat</code>
<code>onmediacomplete</code>	erzeugt wenn das Medium zum Element (Media-Element) komplett geladen wurde für Animation per Timeline Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline siehe <code>onmediaerror</code>
<code>onmediaerror</code>	erzeugt wenn das Medium zum Element (Media-Element) fehlerhaft geladen wurde oder das Medium einen Fehler enthält für Animation per Timeline Media-Element benötigt vor seiner Animation den Datenfluss des Mediums aus einer Datei (streaming media file) z.B. Gif-Bild, Video etc. sinnvoll für Start des Elementes auf der Timeline ersetzt das Ereignis <code>onmedialoadfailed</code> , das deprecated ist und nicht mehr verwendet



	werden darf !
	siehe onmediacomplete
onoutofsync	erzeugt wenn das Element seine Timeline verliert, also nicht mehr auf seiner Timeline animiert wird (nicht synchron zur Timeline des Elementes) sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked": Eventhandler sollte neu synchronisieren z.B. Reset des Elementes durch .resetElement()
	siehe onsyncstored
onpause	erzeugt wenn das aktive Element auf der aktiven Timeline gerade beginnt zu pausieren auch bei body Objekt
onrepeat	erzeugt mit Start jeder Wiederholung der Animation des aktiven Elementes nicht erzeugt beim Start des aktiven Elementes (siehe onbegin), also des ersten Durchlaufes, der keine Wiederholung ist nicht erzeugt durch Kind des Elementes, wenn Kind nicht selbst einen Handler für onrepeat kodiert hat (kein Heraufreichen des Ereignisses onrepeat zum Elternelement) .repeatCount bzw. .repeat muss > 1 sein: onrepeat wird also .repeatCount - 1 mal erzeugt Kind eines Elementes
onreset	siehe onend, onbegin, .repeatCount und .repeat erzeugt wenn Element zurückgesetzt wurde per Methode .resetElement(), also die Timeline des Elementes den aktuellen Wert laut .begin wieder erreicht und damit zurückgesetzt wurde also nicht beim Initialisieren des Elementes und seiner Timeline durch Instanziierung des Elementes
onresume	erzeugt wenn die Pause eines aktiven Elementes, das auf der Timeline pausiert, beendet wird auch für body Objekt
onreverse	erzeugt wenn das aktive Element rückwärts auf der Timeline animiert wird (auch wenn es sich um eine Wiederholung handelt, also .repeatCount > 0 ist) .autoReverse muss auf "true" stehen
onseek	erzeugt wenn eine Seek-Methode zum Element aktiviert wurde: .seekActiveTime() .seekSegmentTime() .seekTo() .seekToFrame()
onsyncstored	erzeugt wenn Element gerade mit seiner Timeline synchronisiert wird nach vorausgegangenem Abbruch der Synchronisation siehe onoutofsync
ontrackchange	sinnvoll nur bei Zwangssynchronisierung per .syncBehavior auf "locked" erzeugt wenn der Track einer Advanced Stream Redirector (ASX) Datei (*.asx) in der Playliste gewechselt wurde

4.3.2.2.4.3.39.24. Userdaten verwalten (.style.userData Behavior des Internet Explorer)

Userdaten per Datencache (UserData Store) dauerhaft oder befristet verwalten (**Achtung: Missbrauch möglich !**)
ab IE 5.x

Alternative : siehe Collection document.cookie

Umfang des Datenbereiches

Security Zone	Document Limit (KB)	Domain Limit (KB)
Local Machine	128	1024
Intranet	512	10240
Trusted Sites	128	1024
Internet	128	1024
Restricted	64	640

pro Objekt 1 Cache

Achtung: Cache	überlebt das Schliessen einer Browserinstanz kann durch andere Browserinstanz benutzt werden
alle Cachedaten	werden in einem gemeinsamen Verzeichnis gespeichert mit gemeinsamen Protokoll
Cache nicht manipulierbar möglich für Objekte	HTML HEAD TITLE STYLE
Cache muss im selben Ordner liegen wie Webpage	

Bsp.: Wenn Cache im Ordner /private/
und Web page im Ordner /public/
so kann Webpage nicht auf den Cache zugreifen.

Verfallszeitpunkt als Zeitstempel für gecachte Daten:
per Eigenschaft .expires lesbar und setzbar



Daten werden **automatisch** durch den Browser gelöscht, wenn das Datum der Daten anhand des Zeitstempels als verfallen erkannt wurde, also der aktuelle Zeitpunkt laut PC-Uhr jünger ist, als der Zeitstempel.

Umwandlung in UTC-Zeitformat per Methode .toUTCString()

Syntax:

```
XML    <Prefix: CustomTag ID=kette STYLE="behavior:url('#default#userData')" >
HTML   <ELEMENT STYLE="behavior:url('#default#userData')" ID= kette >
Scripting object.style.behavior = "url('#default#userData')"
        object.addBehavior ("#default#userData")
```

Hinweis zur XML-Kodierung:

Anstelle der Tagbegrenzung>

sollte ein eigenständiges Endetag kodiert werden, also

.....>

....

</.....>

Prefix	laut XML-Namensraum
CustomTag	User-definierter Tag
kette	String mit dem ID muss kodiert werden

Beispiel:

```
<HTML>
<HEAD>
<STYLE>
    .user_data_speicher_klasse {behavior:url(#default#userData);}
</STYLE>
<SCRIPT>
    // Cachename festlegen
    //      es können diverse Cachennamen definiert und somit Versionen von Cache
    //      verwaltet werden
    var FreierCacheName = "InputCache";

    // Cache-Attribut festlegen
    //      es können diverse Attribute definiert und somit Versionen von Input-Daten
    //      verwaltet werden
    var FreiesCacheAttribut = "InputCacheAttribut";

    // zu cachende Daten referenzieren
    var InputDatenObjekt = ID_Formular.ID_Input;

    function InputSichern()
    {
        // ++++++++ Zeitstempel ++++++++
        var ZeitpunktJetzt = new Date();
        var ZeitpunktJetztInMinuten = ZeitpunktJetzt.getMinutes();

        // Zeitstempel festlegen: Ab jetzt + 20 Minuten
        var Zeitstempel = ZeitpunktJetztInMinuten + 20;

        // und als UTC-Format erzeugen
        var ZeitstempelUTC = Zeitstempel.toUTCString();

        // und Zeitstempel dem Input-Objekt verpassen
        ID_Input.expires = ZeitstempelUTC;

        // ++++++++ Daten chachen ++++++++
        // aktuelle Daten holen laut Input-Objekt
        var InputDaten = InputDatenObjekt.value;

        // Attribut instanzieren und mit Daten füllen
        InputDatenObjekt.setAttribute(FreiesCacheAttribut, InputDaten);

        // und Cache saveen
        InputDatenObjekt.save(FreierCacheName);
    }

    function InputLaden()
    {
        // Cache laden
        InputDatenObjekt.load(FreierCacheName);
```



```

        // und Daten zum Attribut laut Sicherung lesen
        var InputDaten = InputDatenObjekt.getAttribute(FreiesCacheAttribut);
    }

</SCRIPT>
</HEAD>
<BODY>
    <FORM ID="ID_Formular">
        <INPUT ID="ID_Input"
            CLASS="user_data_speicher_klasse"
            TYPE="text"
        >
        <INPUT TYPE="button" VALUE="sichern der Input-Daten" onclick="InputSichern()">
        <INPUT TYPE="button" VALUE="laden der Input-Daten" onclick="InputLaden()">
    </FORM>
</BODY>
</HTML>

```

Eigenschaften:

.expires Zeitstempel der Daten im UserDataStore (User-Cache) per .style.userData Behavior
 Zeitstempel als Verfallszeitpunkt für automatische Löschung gecachter Daten:
 Daten werden **automatisch** durch den Browser gelöscht, wenn das Datum der Daten anhand des
 Zeitstempels als verfallen erkannt wurde, also der aktuelle Zeitpunkt laut PC-Uhr jünger ist, als
 der Zeitstempel.
 String im UTC (Universal Time Coordinate)-Format

Methoden:

.load() gespeicherten UserDataStore (User-Cache) auslesen per .style.userData Behavior
 Cache muss per Methode .save() zum Behavior gespeichert worden sein
.save() UserDataStore (User-Cache) speichern per .style.userData Behavior

4.3.2.2.4.3.39.25. HTML-Dokument mit Style-Sheet (CSS) im IE und NS

Nachfolgende Beschreibung ist die **HTML-konforme** Erweiterung des Objekt-Eigenschaft .style.

Aufgrund der HTML-Integration von CSS sollte der Browser CSS unterstützen. Ob diese Unterstützung vollständig ist, hängt vom Willen des Browserherstellers und seiner am Markt angebotenen Browserversionen ab. Dabei gilt: Auch wenn der HTML-Standard Style-Sheets implementiert hat und ab HTML 4.01 diverse Tags und Attribute als deprecated (missbilligt) setzt, weil sie vereinheitlicht durch genormte Stylesheets zu ersetzen sind, kann der Browser CSS nur im Umfang der implementierten Objekt-Eigenschaften von .style unterstützen. Wie immer gilt: Probieren geht über Studieren.

Microsoft setzt verstärkt auf XML, dessen Komponenten auf Basis der browsereigenen Objektimplementation mehr oder weniger HTML und CSS unterstützen. XML ist die perfektere Abbildung der browser-internen Objektstruktur und somit eine intern komplett objektorientierte Scriptsprache, allerdings hersteller- und browserspezifisch. Letztendlich wird CSS sinnigerweise als grundlegende Komponente und Eigenschaft **aller** Objekte integriert. Ein Vorteil, den Script-Programmierer beim Internet Explorer zu schätzen (lernen) wissen und den andere Browserhersteller mehr und mehr ihren Usern anbieten. CSS und .style bieten elementare Vielfalt bei der dynamischen Scriptprogrammierung, sind gewöhnungsbedürftig aber durch ihre Normung universell anwendbar.

Es sei darauf hingewiesen, dass die Verwendung von CSS-Klassen, Kodierung von <STYLE> bzw. dem STYLE-Attribut bzw. der Eigenschaften.style alle synonym sind. Vier Formen für dieselbe Sache, aber mit folgenden Unterschieden: CSS-Klassen können dokumentweit verwendet werden; <STYLE>, STYLE-Attribut und .style sind HTML-komponentenbezogen.

Hinweis: Margin: Abstand des Aussenrandes eines Objektes zur Umgebung
 Padding: Abstand des Objekthinhaltes zum Aussenrand

Bsp: zu .style: Es sollen DIVs erzeugt werden und deren Objektreferenzen (Zeiger) in einem
 Zeigerfeld gesammelt werden. Das Zeigerfeld dient der vereinfachten
 Verwaltung der dynamisch erzeugten DIV's.

```

var DIV_ID="Test_DIV"; // auch "Otto_DIV" etc.möglich , je nach Wunsch

var DIV_Zeiger=Array(); // sammelt alle ID als Zeichenketten

// DIV's dynamisch erzeugen und Zeiger einsammeln
var Left= 20; // Abstand vom linken Fensterrand in Pixeln
for (var i=0; i < 3; i++)
{
    Left+=10; // ab 30 Pixel vom linken Fensterrand
    document.write(
        '<DIV ID="' + DIV_ID + i.toString() + "' '
        + ' STYLE="position:absolute;'
        + 'left=' + Left + 'px;'
        + 'top=20px;'
        + '""
        + '>'
        + '</DIV>'
    );
    eval ('DIV_Zeiger[' + i + ']' += DIV_ID + i.toString() );
}

```

